A special theory for the generation and recognition of symmetric messages in cybernetic systems.

Hayes, D

101

A SPECIAL THEORY FOR THE GENERATION AND RECOGNITION OF SYMMETRIC MESSAGES IN CYBERNETIC SYSTEMS

A thesis submitted for the degree of Doctor of Philosophy in Cybernetics in the Faculty of Science, University of London by

David Hayes
BSc, MPhil, CertEd

Supervisor: Dr H A Fatmi

Department of Electronics
Chelsea College
LONDON SW6 5PR

September 1979

A SPECIAL THEORY FOR THE GENERATION AND RECOGNITION OF SYMMETRIC MESSAGES IN CYBERNETIC SYSTEMS

ABSTRACT

This work was carried out in the Department of Electronics, Chelsea College, University of London, under the supervision of Dr H A Fatmi.

The thesis proposes a special theory for the generation and recognition of symmetric messages within the context of 1-D signals and 2-D patterns.

Particular attention is paid to extending the existing definitions of bilateral symmetry to local regions of 1-D discretely sampled and continuous real spaces. Original definitions are proposed for 2-D discrete binary spaces that rely on the contour extraction technique of "vector chain mapping" previously developed by the author.

Based on these definitions of symmetry, integral error functions that measure deviations away from perfect symmetry are derived. These are shown to be of special application in areas of signal processing and pattern recognition and in themselves lead to further detection criteria.

In order to assess the performance of the detection techniques for symmetric messages, general association functions are introduced within the context of a Cybernetic system - conceptualised to consist of a pattern generator, a pattern recogniser and an optimiser/controller.

This leads to the problem of choosing objective functions on which to optimise the parameters within the symmetric detection process. The sequential simplex technique is applied to a carefully chosen statistical performance criterion that relates to a simulated radar system.

The thesis concludes by considering what remains unanswered. Ways of extending the special theory to self- and mutually-referential abstract symmetries, using property operators, are indicated.

ACKNOWLEDGEMENTS

I should like to express my gratitude to the Electronics Department at Chelsea College, University of London, for their support of my research; in particular Dr H A Fatmi, whose advice and encouragement have been deeply appreciated.

I also thank my employers - Marconi Avionics - who have allowed me the use of their facilities in producing the two papers included in the appendices.

C O N T E N T S

A LIST OF FIGURES

Figure No | Page No

LIST OF SYMBOLS

Boolean Variable

B	General Boolean Variable

Integer Variables

M, N, N', Q, R, S, X, Y	General Integer Variables
E, H, K	Extents
T, T'	Orders

Real Variables

i, j, m, n, r, s, t, x	General Real Variables
e, h, k	Extents
ρ, e_t, θ, Θ	Thresholds
ϕ	Angle

Vectors

$s_1 \dots s_8$	Ordinal Vectors
v	General Vector (Integer Components)

Symbols

a, b, c	Alphabetic Symbols
$m_1 \dots m_n$	Message Symbols

Discrete Functions

F, F_a, F_b, G	General Discrete Functions
W	Weighting Function
Ψ, Φ_F, Φ_{GF}	Correlation Functions
${}_eS_e$, ${}_oS_e$, ${}_e\mathcal{S}_o$, ${}_o\mathcal{S}_o$, $\mathscr{S}_e$, $\mathscr{S}_e'$	1D Symmetry Functions
Z	Rotational Symmetry Function
Z'	Reflexional Symmetry Function

Discrete Functions (cont'd)

D	Detection Function
Δ $\blacktriangle$	Decision Functions
A A' A_c A_c'	Association Functions
I	Generation Function
$\mathcal{J}$	Recognition Function
O	Objective Function
O_s	Stochastic Objective Function

Continuous Functions

f, g	General Continuous Functions
$\mathcal{E}_e$	Error Function
s_e, s_e', s_o, s_o'	Symmetry Error Functions
$\emptyset, \emptyset_f, \emptyset_{gf}$	Correlation Functions

Probability Functions

p_s	Short term 'a posteriori' probability of success
p_e	Short term 'a posteriori' probability of error
P_s	Long term 'a posteriori' probability of success
P_e	Long term 'a posterior' probability of error

Matrices

$R_{\emptyset}$	Rotation Matrix
$R'_{\emptyset}$	Reflexion Matrix
C	Cellular Boolean Matrix

Vector Chains

V	Vector Chain of Ordinal Composition
V'	Vector Chain of General Composition

Sets

Rl	Set of Real Numbers
In	Set of Integer Numbers

Equivalences

$=$	Numerical Equivalence
$\underset{s}{\sim}$	Symmetric Equivalence

Operators

$\sum$	Summation Operator
$\int$	Integration Operator
$\mid\ \mid$	Modulus Operator (eg $\mid -5 \mid = 5$)
$\lceil\ \rceil$	Truth Operator (eg $\lceil 3>5 \rceil = 0$, $\lceil 2^3=8 \rceil = 1$)
B	Binary Remainder Operator (eg B(5)=1,B(8)=0)
P	Absolute Property Operator
ϱ	Relative Property Operator
ϱ^{-1}	Inverse Relative Property Operator
θ_r	Modulo Shift Operator (eg $\theta_2(1,2,3,4)=(3,4,1,2)$)
$\not{s}$	Production Operator

* Also referenced as Appendix Two
** Also referenced as Appendix Four

Tyger! Tyger! burning bright
In the forests of the night,
What immortal hand or eye
Dare frame thy fearful symmetry?

the last stanze of The Tyger (Experience)

by William Blake

INTRODUCTION

> "Symmetry is not merely a descriptive nicety; like other thoughts in Pythagoras, it penetrates to the harmony of nature."
>
> from: 'The Music of the Spheres' by Jacob Bronowski

When one looks at a snowflake one sees symmetry that approaches perfection, until it begins to melt. Similarly, when one views the cross-sectional motion of a fluid that has had a pebble dropped onto its surface, one initially sees a symmetric wave pattern that will persist until the edges of the containing vessel and the frictional properties of the fluid cause the rippling motion to decay into asymmetry. The physical examples of symmetry of this sort seem almost endless, but, how do they come about and how does one recognise what is essentially an abstract mathematical idea as a physical reality in nature? Cybernetics is concerned with situations of high order, that is, pockets of "neg-entropy". Such phenomena that display symmetric properties are indicative of such order. That is to say, there exist a transition sequence embedded in the matter associated with the phenomenon that gives rise to the symmetric messages one intercepts - deviations in symmetry suggesting non-homogeneity in the surrounding medium.

Even the statistical nature of most natural events is described by the normal distribution, into which all other distributions merge when averaged, which is itself symmetric about its mean value. Thus there exists a fundamental tendency in nature to work within the bounds of symmetry - every particle when viewed in isolation is postulated to have symmetric electric, magnetic and gravitational fields. How many arguments in physics rest on the phrase: 'by considering symmetry'?

Hence, this thesis is to discuss the generation and recognition of symmetric messages. In so doing it shall help unite two central areas of Cybernetics, namely: signal processing and pattern recognition, as well as making use of such integral techniques as system modelling, optimisation theory and decision theory.

In general a message may either be viewed as a signal or pattern. A signal is here restricted to mean a one dimensional temporal sequence of information and a pattern is restricted to a two dimensional spatial array of information - this is a matter of convenience in terminology and is a totally arbitrary distinction that may be ignored outside the confines of this text. Mathematically a discrete signal is represented as a vector and a pattern a matrix. Clearly a message may combine the properties of both a signal and a pattern, by the use of higher order matrices.

Particular emphasis in this text shall be placed on developing a clear theoretical basis, that may be applied to real-life situations, which can be identified and simulated to demonstrate the effectiveness of the developed techniques, and thus used to optimise the performance of the considered generation and recognition processes, according to some 'ad hoc' criterion of goodness.

The initial impetus to begin this work began with the simple realisation that when an energy source is scanned across by a transducer with a spatially symmetric response characteristic (such as a microphone or an antenna) then a symmetric signal is generated in the channel from the transducer - provided certain physical conditions regarding the energy source (eg constant power) and the scanning rate (eg linear) are satisfied. On investigation no special theory was found to exist to cover the detection of the class of symmetric signals in either the theory of signal processing (ie the separation of signal from noise) or the theory of pattern recognition (ie the classification of patterns). It was therefore decided to adapt and develop the present theories from both areas and study the resulting coalition under the important restriction of symmetry.

Soon after commencement it became obvious that certain fascinating properties of systems, inverse systems and meta-systems were going to emerge from further study and ideas initially pertaining to one dimensional signals could be extended to two dimensional chains, given the technique, previously developed by the author, of vector chain mapping (Hayes 1, 1976). It was through these fortunate realisations that the writer gained the enthusiasm to produce this work on the generation and recognition of symmetric messages in cybernetic systems.

Thus the first chapter defines the nature of a symmetric message for both the continuous and discrete cases in one dimension and in two dimensions for only the discrete case. Although this chapter is in parts fairly elementary, it is essential to the careful development of the identification theory and does allow opportunity to discuss some interesting examples of how symmetric signals and patterns are generated.

The second chapter introduces the notion of measuring deviations in symmetry based on the definitions given in chapter one and also compares methods of both traditional and novel lines for the identifying of the axes of symmetry of a noisy symmetric signal or pattern.

In the third chapter, the idea of a cybernetic system is introduced and this is used in a computer-aided design exercise for the generation of symmetric signals met in a radar environment and then for simulating the detection process from which simple statistics can be automatically established to appraise the overall efficiency of the recognition process.

Penultimately, the fourth chapter extends chapter three by applying optimisation techniques to the symmetric message detection processes by introducing the well-established idea of an objective function and basing it on the probabilities of error and success.

Finally, some conclusions are drawn concerning the directions of future development of the special theory and its application to other areas in cybernetics.

Also intrinsic to the thesis are four appendixes. The first is a paper by the author on the application of symmetric detection theory to a radar system. The second is a computer program that performs the computer-aided design mentioned previously. The third is a paper on a cybernetics of seeing that extends the ideas developed in this text to the real world. The fourth is a computer program to map a two dimensional binary matrix into a set of vector chains. All are included to give added credence to the techniques described.

Before entering the main text, it is important for the reader to realise that in the main Cyberneticians have previously, largely ignored the concept of symmetry, with only oblique reference to the idea being made by Wiener in his authoritative book Cybernetics (Wiener (1), 1948). For example, in his chapters on Gestalt and Universals and Groups and Statistical Mechanics, where such phenomena as rotations, translations and equilibria are discussed, the word 'symmetry' is never used! Likewise Ashby makes no direct mention of the term when discussing change and stability (Ashby 1956). Minsky on the other hand does discuss palindromes in his treatise on computation (Minsky, 1967) and with Papert discusses symmetry along a line in the context of a perceptron (Minsky and Papert, 1969). However, they do not generalise this idea to two dimensions.

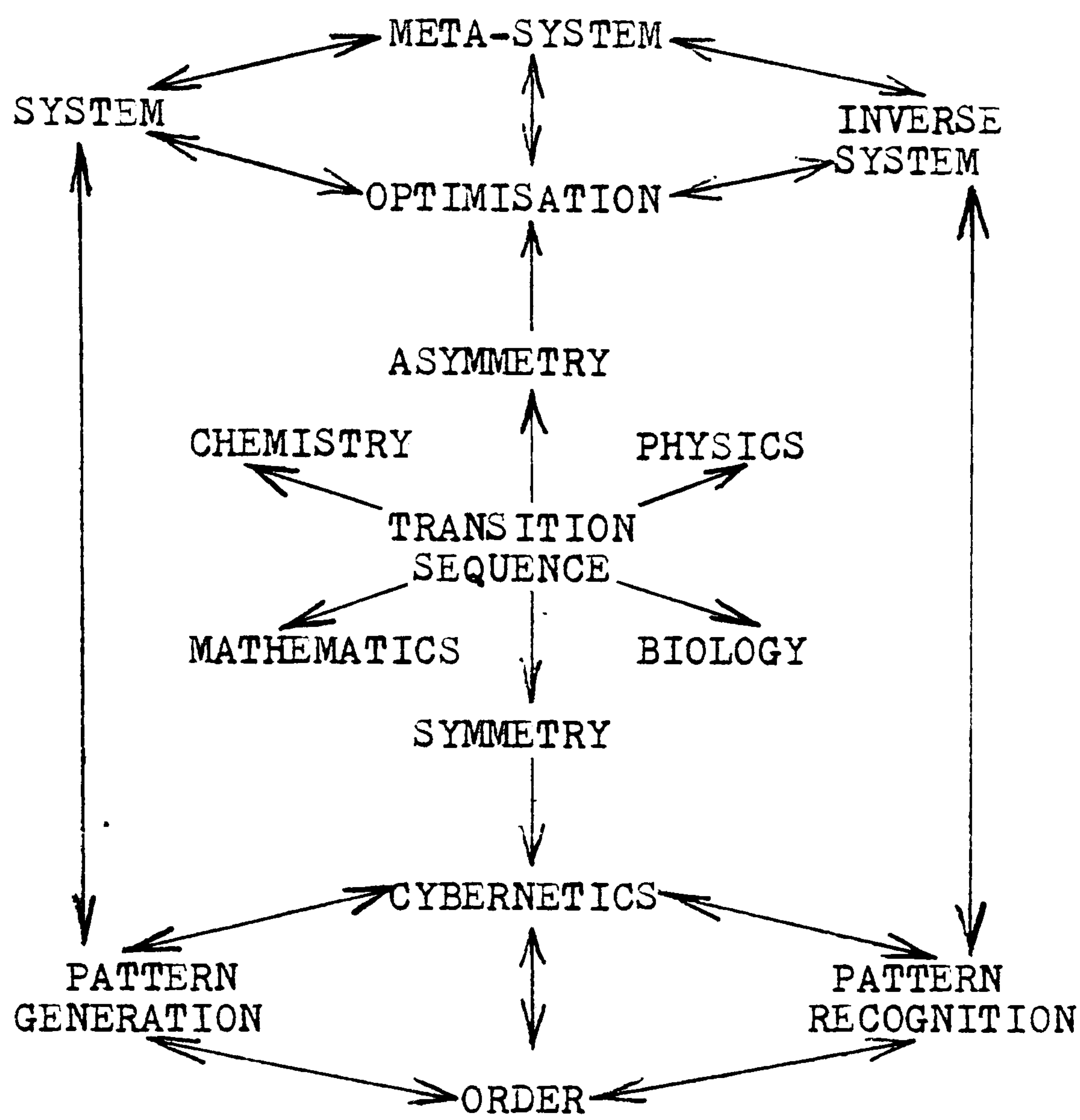
The author finds these oversights in the foundation of Cybernetics curious, for, as the mathematician Hermann Weyl concludes:

> "Symmetry is a vast subject significant in art and nature."
>
> Weyl, 1952

Also Weyl identifies such eminent figures as Leibniz, Leonardo da Vinci, Hermite, Galois, Mach and Plato with the concept of symmetry, all of whom Wiener also related to Cybernetics.

It is therefore hoped that the following thesis will go some way towards making the appropriate links between Cybernetics and symmetry. However, a rigorous foundation for Cybernetics based on symmetry considerations is beyond the scope of this text. Further to this, it is emphasised that, although Weyl's work is an erudite and totally recommended study, it only just begins to answer the central question of this thesis: "How is symmetry recognised?" At little appears to have been written on the general concept of symmetry since Weyl, much of what follows has had to be worked out from first principles and is therefore largely standing alone.

To conclude this introduction, an entailment structure for the cybernetic concept of symmetry is given overleaf in Figure 1.



AN ENTAILMENT STRUCTURE FOR SYMMETRY

FIGURE 1

CHAPTER ONE

DEFINITION OF SYMMETRIC MESSAGES

Foreword

This is a definitive chapter on the nature of symmetric messages. It deals with the generation of symmetric signals and makes some observations on the conservation of symmetry under the convolution operation. This is followed by a mathematical definition of symmetry along a line, dealing in particular with the additive interference of two symmetric signals.

Next the generation of a symmetric pattern is discussed and different types of symmetry (actually extents) are defined, relating multistate patterns with equipotential boundaries.

1.1 The Class of Locally Symmetric Signals

"Pleon hemisy pantos" *

Hesiod, Erga, 40

This section deals with the mathematical definition of a symmetric signal, but it is instructive to first consider the accepted meaning of the word 'symmetry'. It is derived from the Greek 'symmetria' which has the roots: 'syn' interpreted as 'together' and 'metron' interpreted as 'a measure'. In English it takes the meaning of: 'exact correspondence of parts on either side of a straight line or plane, or about an axis or centre: balance or due proportion: beauty of form: disposition of parts' (from the Chambers Dictionary).** Clearly this gives plenty of scope to the mathematical definition, but, before discussing symmetric signals in a totally abstract way, it is interesting to consider certain physical realisations of symmetry and study how these have come about.

1.1a The Generation of Symmetric Signals

As was explained in the introduction, there are many ways symmetric signals may arise; for example, a raindrop falling on the surface of a calm pool of water or a beam of light being diffracted by a small symmetric aperture or a tiger evolving the same camouflage on both its sides. In the next three sub-sections, three different ways of generating symmetric signals will be discussed and some important conditions for the conservation of symmetry will be indicated.

* Translation: The half is more than the whole.

** 'Ebenmass' is a good German equivalent for the Greek symmetry: for like this it carries also the connotation of "middle measure"

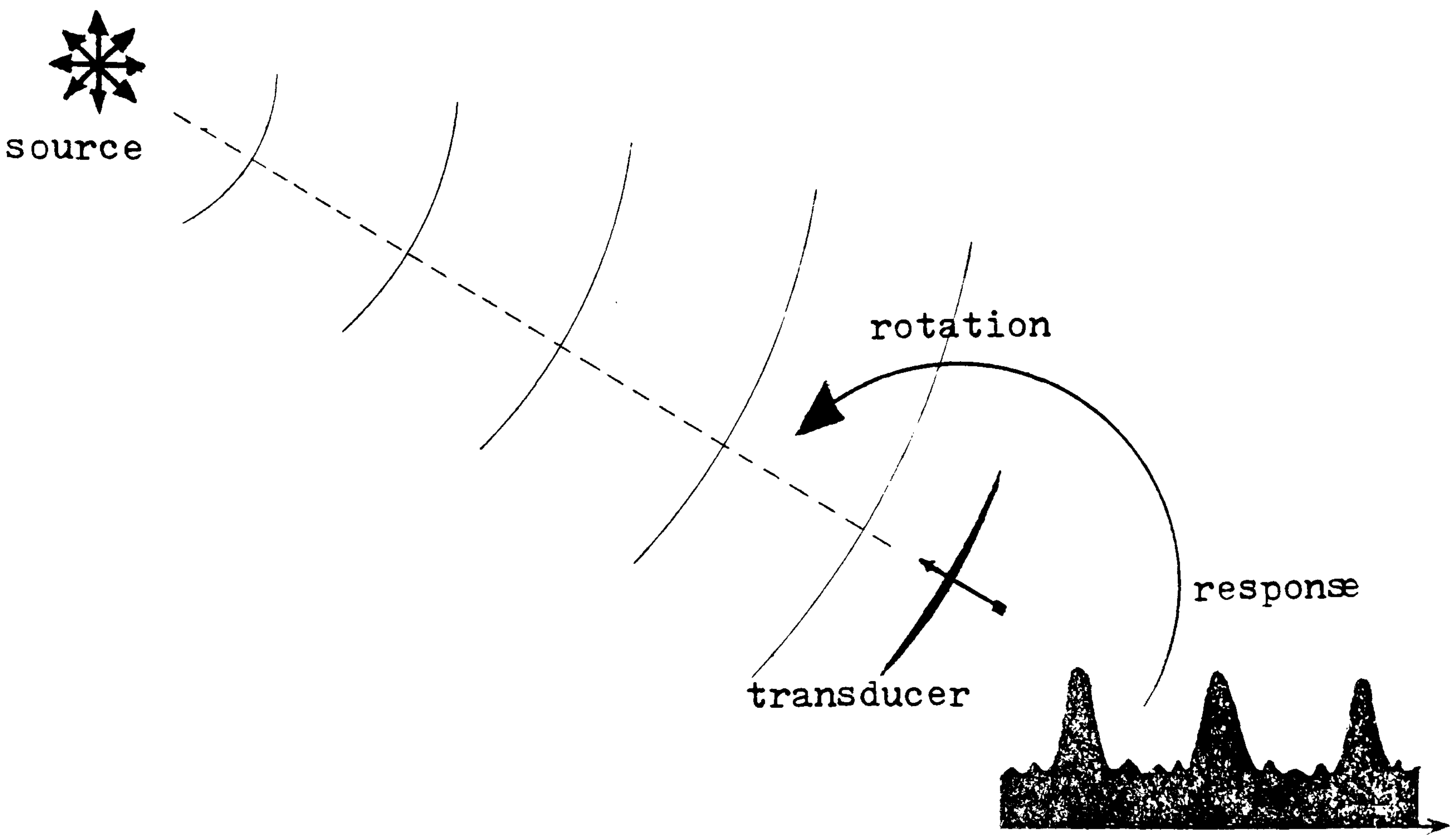
Weyl, 1952

1.1.a.i A stationary source with a rotating transducer

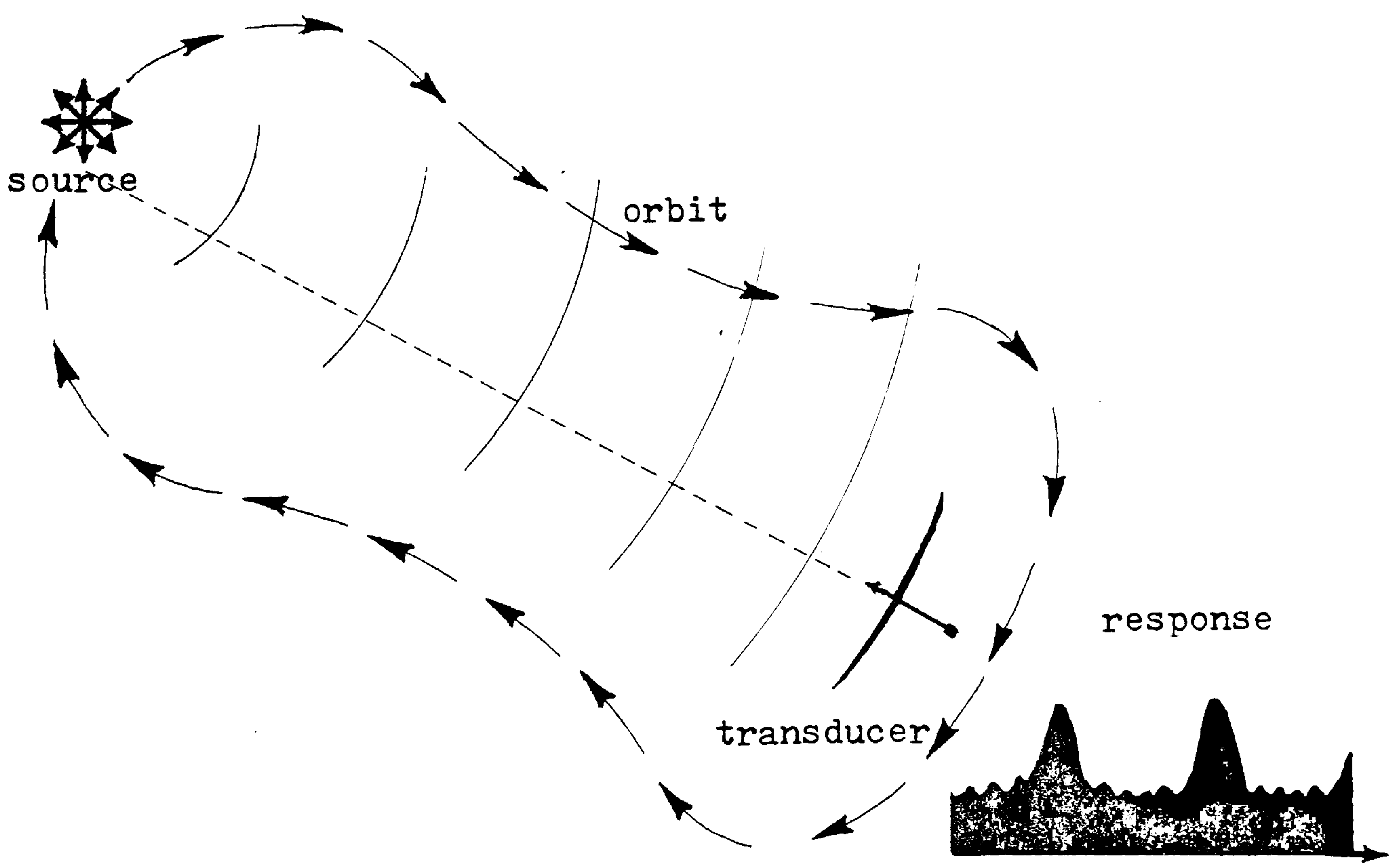
Consider a solitary stationary source emitting a continuous signal of constant energy. If a transducer that is attuned to this source is rotated at a steady rate about a fixed axis as in Figure 1.1, then the output of the transducer represents the spatial characteristics of the transducer, which in most cases will be approximately symmetric due to the spatially symmetric construction of the device itself. A realistic example is a rotating antenna receiving returns from a constant energy source, which may either be illuminated or self excited - as are the cases of radar systems and direction finding equipment respectively. Plainly a wide range of possibilities exist under these circumstances concerning the exact nature of the energy source (eg pulsed etc), the transducer (eg sampled etc), and the relative orientations, rates and angles of scan, but provided some matching is performed (ie synchronisation), then a symmetric response will result.

1.1.a.ii An orbiting source with a stationary transducer

The converse of the situation just considered is to have a fixed transducer (ie not rotating) orbited by a constant uniform energy source at a fixed radius. In fact a more general situation may be envisaged where the orbit is non-circular and need only be symmetric about the axes of symmetry of the transducer, as is shown in Figure 1.2, to give a symmetric response.



STATIONARY SOURCE - ROTATING TRANSDUCER FIGURE 1.1



ORBITING SOURCE - STATIONARY TRANSDUCER FIGURE 1.2

1.1.a.iii Electronic realisations

A transversal filter with a symmetric weighting sequence as is shown in Figure 1.3, is another way of producing or simulating a symmetric signal. The Nth response G(N) of such a system is:

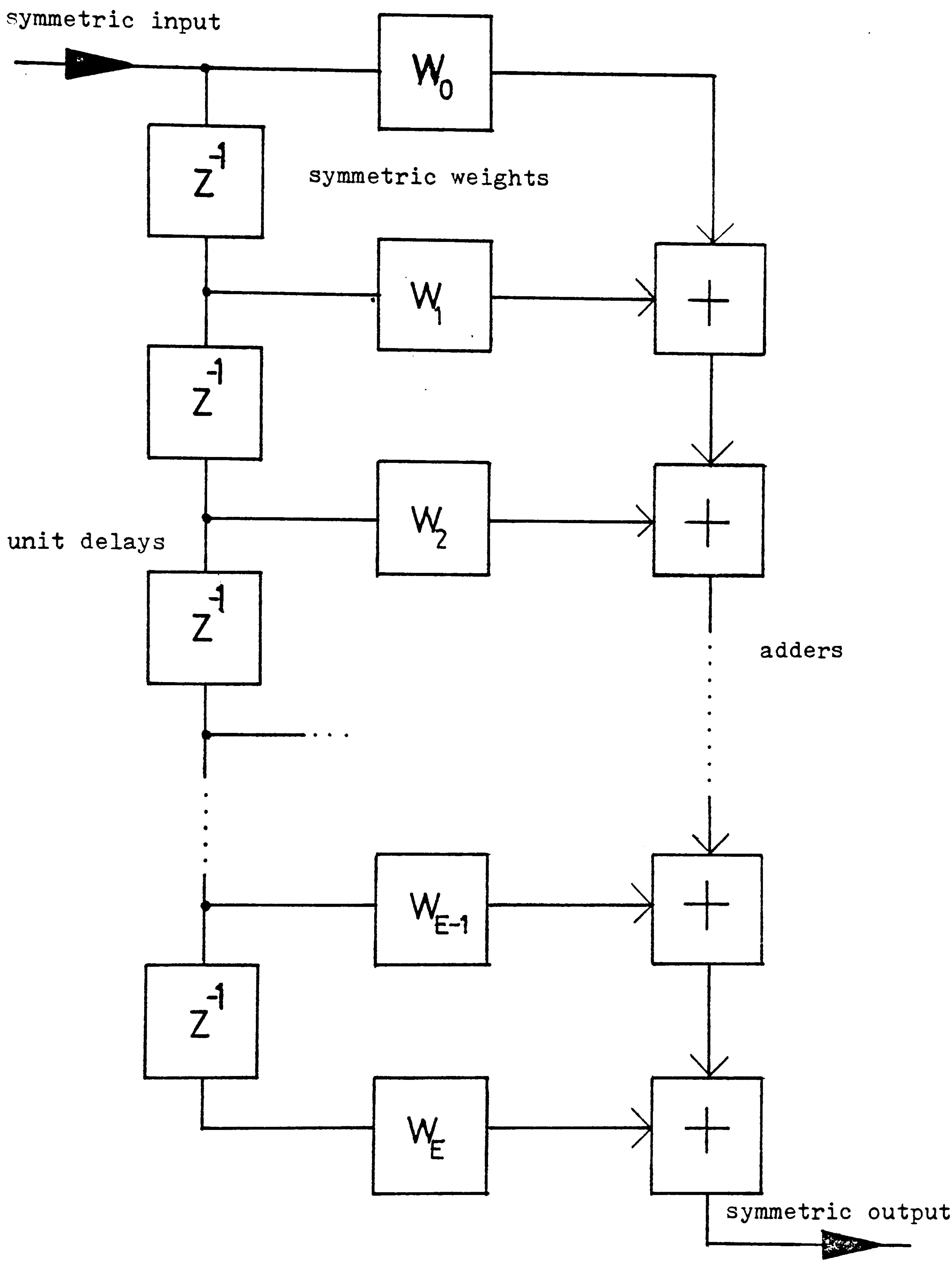
$$G(N) = \sum_{R=0}^{R=E} W(R).F(N-R) \tag{1.1}$$

where F(N) is the Nth input, W(R) the Rth weight and E is the order of the filter.

Clearly expression 1.1 is a convolution* and will produce a symmetric response only if the input signal is symmetric (eg an impulse).

Thus symmetry maintains symmetry and all symmetric transducers require symmetric situations (inputs) to produce symmetric outcomes (outputs), because the transducers must perform convolutional operations of the form of expression 1.1. That is to say that the physical mappings between time and space conserve symmetry only if these mappings are themselves symmetric in every respect. For instance, the rate of rotation in the example given in sub-section 1.1 a.i may be non-linear but must be symmetric and the velocity, acceleration and higher order changes in the orbiting energy source in the example given in sub-section 1.1.a.ii must also be symmetric about the axis of symmetry of the transducer. Hence symmetric signals are delicate commodities that must be treated carefully if they are to be preserved.

* for a good description of the convolution process see Brown (1965)



A TRANSVERSAL DIGITAL FILTER   FIGURE 1.3

However, it is possible to generate symmetry from an asymmetric situation by applying the auto-correlation function Ψ to a discrete signal F, that is:

$$\Psi(N) = \mathop{\mathcal{L}t}_{E\to\infty} {}^{1}/_{2E} \sum_{R=-E}^{R=E} F(R).F(N+R) \tag{1.2}$$

which is equivalent to expression 1.1 except W(R) has been replaced by F(R) and F(N-R) has been sequence reversed to become F(N+R); also the limits have been extended to infinity and the sum normalised. Hence convolution is identical to auto-correlation when W(R)=F(R) and F(R) is symmetric (eg F(N+R)= F(N-R) for all R), the limits being appropriately adjusted. Correlation is identical to convolution if F is symmetric. Lastly, it should be noted that the information content of a symmetric signal is redundant by approximately a factor of 2, only adding 1bit more information to that contained in half the signal.

1.1.b Some Formal Definitions of Locally Symmetric Signals

To this point it has been implicit that the reader has understood exactly what is meant by a symmetric signal. This can no longer be the case for even the simplest concepts are subject to varied interpretations and it is therefore necessary to turn to some established mathematical ideas to agree on the formal definitions of symmetry.

The concepts of odd and even symmetry are well known and these will now be extended to apply to local regions in continuous and discrete spaces. The major emphasis shall be on even symmetry, since this type of symmetry will be made more use of in successive chapters.

Hence, the following text develops the properties of local symmetry for discrete and continuous, one dimensional signals, $F(X)$ and $f(x)$ respectively, where X belongs to the set of integers and x the set of real numbers.

1.1.b.i An even, continuous local symmetry

$f(x)$ is an even locally symmetric function, iff, for some real number m:

$$f(m-r)=f(m+r) \text{ for all } r, \text{ such that } 0 < r \leqslant k; \tag{1.3}$$

where m is a pivoting point called "the centre of local symmetry" k is known as "the extent of local symmetry" and $2k$ is the range of local symmetry.

1.1.b.ii An even, discrete local symmetry

$F(X)$ is an even locally symmetric function of even composition, iff, for some integer number M:

$$F(M-R)=F(M+R+1) \text{ for all } R, \text{ such that } 0 \leqslant R \leqslant K; \tag{1.4}$$

where M, R, K and $2K$ are defined in the same way as their continuous counterparts: m, r, k and $2k$.

$F(X)$ is an even locally symmetric function of odd composition, iff, for some integer number M:

$$F(M-R)=F(M+R) \text{ for all } R, \text{ such that } 1 \leqslant R \leqslant K \tag{1.5}$$

where M, R and K are as before, but 2K+1 is now the range of symmetry.

Note that there are two types of discrete symmetry along a line and the centre of local symmetry for the evenly composed case is a matter of arbitrary choice to the nearest integer step. However, if fractional concepts are allowed, then the centre may be defined at $M+\frac{1}{2}$.

1.1.b.iii Some observations on odd and global symmetry

f(x) is an odd locally symmetric function, iff, for some real number m:

$$f(m-r)=2f(m) - f(m+r) \text{ for all } r, \text{ such that } 0 \leqslant r \leqslant k; \tag{1.6}$$

where m, r, k and 2k are as defined in sub-section 1.1.b.i. F(X) is an odd locally symmetric function of even composition, iff, for some integer M:

$$F(M-R)=F(M)+F(M+1)-F(M+R+1) \text{ for all } R, \text{ such that } 1 \leqslant R \leqslant K \tag{1.7}$$

where M, R, K and 2K are as defined for the evenly composed discrete symmetry.

F(X) is an odd locally symmetric function of odd composition, iff, for some integer M:

$$F(M-R)=2F(M) - F(M+R) \text{ for all } R, \text{ such that } 1 \leqslant R \leqslant K \tag{1.8}$$

and the range of local symmetry is now 2K+1.

The above definitions are given for completeness and will not be used in further analysis (excepting Chapter 2).

Observe that the properties of local symmetry for both the odd and even cases for the continuous functions follow directly from the conventional cases of global symmetry about the origin in pure mathematics but, as will have been noticed, greater care is needed with discrete functions where, as has already been indicated, for the even case, the centre of local symmetry is a matter of arbitrary distinction to within an integer step - this is clear if $F(X)$ is thought of as a set of regularly sampled values of $f(x)$, and as the sampling interval decreases the limiting centre of local symmetry can be approached from either side.

If either x or X is bounded, then all previous definitions are still true, but care must be taken to ensure edge effects are accounted for by defining, for example, the functions f or F to be cyclic (eg X is modulo N - where N is finite and integer). Concepts of global symmetry may be arrived at by allowing k and K to approach infinity in the non-modulo case.

1.1.c The Decomposition of Two Compounded Symmetric Signals

A real possibility is that two symmetric signals may combine to form an asymmetric composite, as is the case when two angularly separated sources are scanned across by an antenna. If this occurs, it is possible to separate these two signals given the sparse information that they are of equal extent and do not entirely overlap.

The exact applications for the following theory are not as yet certain, but it may be of use in transmitting overlapping fixed length multi-level codes, where a high degree of mixing is desirable. Only the discrete case will be considered, but basically the technique is one searching for the beginning of the first message. It is then known where it terminates and all that follows must be the conclusion of the second message. Clearly this may be folded back on itself over its extent and subtracted from the composite message. This process can then continue until both messages are separated.

Consider the problem of, given any discrete function F, of known dimension T, deciding whether or not it can be decomposed into two symmetric functions F_a and F_b (see sub-section 1.1.b.ii) of equal and known dimension T', where $T' \leqslant T$. By representing F, F_a and F_b by vectors, it follows that:

$$F=(F_a,\underbrace{0,0,\ldots 0)+(0,0,\ldots 0}_{\text{R zeroes}},F_b) \qquad (1.9)$$

where R is defined as the shift and is related to T and T' by:

$$R=T-T' \qquad (1.9a)$$

Further to this, if expression 1.9 is re-expressed in terms of its components:

$$(F(1)\ldots\ldots F(T))=(F_a(1)\ldots\ldots F_a(1),0,\ldots 0)+(0,0,\ldots 0,F_b(1)\ldots F_b(1)) \qquad (1.10)$$

and by equating components the following set of T independent equations with T'+2 unknowns are formed:

$$\left.\begin{array}{l} F(1)=F_a(1)+0 \\ F(2)=F_a(2)+0 \\ \\ F(R)=F_a(R)+0 \end{array}\right\} \quad \text{R equations}$$

$$\left.\begin{array}{l} F(R+1)=F_a(R+1)+F_b(1) \\ F(R+2)=F_a(R+2)+F_b(2) \\ \\ F(t)=F_a(1)+F_b(t-R) \end{array}\right\} \quad \text{T-2R equations}$$

$$\left.\begin{array}{l} F(t+1)=0+F_b(t-R+1) \\ F(t+2)=0+F_b(t-R+2) \\ \\ F(T)=0+F_b(1) \end{array}\right\} \quad \text{R equations} \qquad (1.11)$$

from which the solutions for F_a and F_b must be extracted.

For $N \leqslant R$ the solutions can be seen immediately to be:

$$F_a(N)=F(N) \qquad (1.12a)$$

and by exploiting symmetry:

$$F_b(N)=F(T+1-N) \qquad (1.12b)$$

For $R < N \leqslant 2R$

$$F_a(N)=F(N)-F(T+R+1-N) \qquad (1.13a)$$

and

$$F_b(N)=F(T+1-N)-F(N-R) \qquad (1.13b)$$

by substituting the solutions given in 1.12a and 1.12b into 1.11.

For $N \leqslant 2R$

$$F_a(N)=F(N)-F(T+R+1-N)+F_a(N-2R) \tag{1.14a}$$

and

$$F_b(N)=F(T+1-N)-F(N-R)+F_b(N-2R) \tag{1.14b}$$

by substituting the solutions given in 1.13a and 1.13b into 1.11 and by noting the occurrences of $F_a(N-2R)$ and $F_b(N-2R)$ in their solutions.

By combining 1.12a and b, 1.13a and b, and 1.14a and b, the solution sets can be expressed as:

$$\left\{F_a(N)/N \leqslant T'\right\} \; ; \; \left\{F_b(N)/N \leqslant T'\right\} \tag{1.15}$$

and the two defining equations may be re-written as

$$F_a(N)=F(N)- \lceil N>R \rceil \, F(T+R+1-N)+ \lceil N>2R \rceil \, F_a(N-2R) \tag{1.16a}$$

and

$$F_b(N)=F(T+1-N)- \lceil N>R \rceil \, F(N-R)+ \lceil N>2R \rceil \, F_b(N-2R) \tag{1.16b}$$

where 1.16a and b are <u>independent</u> <u>recursive</u> solutions with predicated coefficients. Thus, if a and b are Boolean variables defined by

$$a= \lceil N>R \rceil \text{ and } b= \lceil N>2R \rceil \; ;$$

then 1.16a and b become:

$$F_a(N)=F(N)-a.F(T+R+1-N)+b.F_a(N-2R) \tag{1.17a}$$

and

$$F_b(N)=F(T+1-N)-a.F(N-R)+b.F_b(N-2R) \tag{1.17b}$$

It will be noted that 1.17a is solved independently of 1.17b and can therefore form a test for correct decomposition by the substitution of F_a and F_b into 1.10 and then checking their correspondence to the known F. Alternatively, the symmetry function may be applied to F_a and F_b, see section 2.2.

To illustrate the application of 1.17a and b consider:

Example One.

if $F=(1,3,7,7,5,1)$

where $T=6$ and $T'=5$

then the shift $R=T-T'=6-5=1$, and

$F_a(1)=F(1)-\lceil 1>1 \rceil\ F(6+1+1-1)+\lceil 1>2 \rceil\ F_a(1-2)$

$=F(1)-0.F(7)+0.F_a(-1)=F(1)=1.$

$F_a(2)=F(2)-\lceil 2>1 \rceil\ F(8-2)+\lceil 2>2 \rceil\ F_a(2-2)$

$=F(2)-1.F(6)+0.F_a(0)=3-1=2$

$F_a(3)=F(3)-\lceil 3>1 \rceil\ F(8-3)+\lceil 3>2 \rceil\ F_a(3-2)$

$=F(3)-1.F(5)+1.F_a(1)=7-5+1=3$

$F_a(4)=F(4)-\lceil 4>1 \rceil\ F(8-4)+\lceil 4>2 \rceil\ F_a(4-2)$

$=F(4)-1.F(4)+1.F_a(2)=7-7+2=2$

and so on, to give

$F_a=(1,2,3,2,1)$ and $F_b=(1,4,5,4,1)$

which are both oddly composed symmetric functions.

From the previous example it may be noticed that the predicates 'a' and 'b' may be dispensed with, by defining F_a and F_b and F to be zero for all arguments outside their specified range.

That is:

$F(N) \in \{Rl\}$ for $0<N\leqslant T$ otherwise $F(N)=0$

$F_a(N) \in \{Rl\}$ for $0<N\leqslant T$ otherwise $F_a(N)=0$

$F_b(N) \in \{Rl\}$ for $0<N\leqslant T$ otherwise $F_b(N)=0$

where $\{Rl\}$ is the set of real numbers and

$$F_a(N)=F(N)-F(T+R+1-N)+F_a(N-2R) \qquad (1.18a)$$

and

$$F_b(N)=F(T+1-N)-F(N-R)+F_b(N-2R) \qquad (1.18b)$$

Now consider the case of:

Example Two.

if $F=(1,2,3,5,12,6,12,5,4,9,1,7)$

where $T=12$ and $T'=8$.

It may easily be computed by the application of 1.18a and b that

$F_a=(1,2,3,5,5,5,3,1)$ and $F_b=(7,1,9,4,4,10,3,7)$

and it will firstly be noticed that neither F_a nor F_b are symmetric functions and secondly that F_b has been given out in a 'reversed' manner, which, of course, in the case of a symmetric function could not be noticed, but is clearly unacceptable for the non symmetric case, as 1.9 should be satisfied at least over the first and last R values.

Thus 1.18b must be reversed, such that

$$F_b(N) = F_b(T'-N+1)$$

thus

$$F_b(N)=F(R+N)-F(T+1-N-2R)+F_b(T+1-N-3R)$$

and therefore the defining equations become:

$$F_a(N)=F(N)-F(T+1-N+R)+F_a(N-2R) \qquad (1.19a)$$

and

$$F_b(N)=F(R+N)-F(T+1-N-2R)+F_b(T+1-N-3R) \qquad (1.19b)^*$$

In practice an error 'e' will always occur in any decomposition and this value 'e' should be below a certain tolerance threshold 'e_t', that is

$$e < e_t$$

Thus e may be defined as:

$$e=\sum_{N=1}^{N=T}\left|F(N)-F_a(N)-F_b(N-R)\right| /_T \qquad (1.20)$$

which can be accumulated for N increasing.

In the case of example one, the error e is zero and in example two the error e is $3/_{12}$.

* Note that expression 1.19b is only valid if $2N > T+1-3R$ and is only completely realisable in the event of $3R+1 > T$. Whereas expression 1.18b is always true, since $0 > -2R$ as $T > T'$, and should be returned to in the circumstances of 1.19b not being realisable, the vector f_b being reversed in a separate operation.

1.2 The Class of Locally Symmetric Patterns

> "In asking what operations will turn a pattern into itself we are discovering the invisible laws that govern our space. There are only certain kinds of symmetries our space can support, not only in man-made patterns, but in the regularities which nature herself imposes on her fundamental, atomic structures."
>
> from: 'The Music of the Spheres' by Jacob Bronowski

1.2.a The Generation of Symmetric Patterns

The concept of symmetry would never have had such a great influence on modern scientific thought if nature had not given man the power of sight, for it is in the three dimensional shapes that are mapped on to the surfaces of man's two dimensional retina that symmetry abounds. And, this observation suggests, two fundamental properties of man and his universe:

a) that symmetry is somehow essential to man's existence,

and

b) the mechanism by which man recognises symmetry must itself have symmetric properties.

For example, from the very small, that is an elementary particle, to the very large, that is a galaxy, symmetry is invariably seen, and in the living world it is manifold - in leaves, in people's faces and bodies, in art and in a cell's RNA and DNA.

But what generates this symmetry?

A natural question with only a natural answer, in that the reply must be the elementary forces of nature and space; (gravity,radiation, electromagnetism and nuclear) all tend to mold man and his environment into symmetric forms because they are themselves symmetric.

This point is discussed at length by Weyl (1952) where he points out that Leibniz argued left and right are absolutely indistinguishable in the physical universe:

> "Just as all points and all directions in space are equivalent so are left and right."

They are simply relative concepts, as are clockwise and anticlockwise. This physical lack of preference seems to suggest that the universe should be totally symmetric.

Manifestly, everything is not totally symmetric. If that were the case, then the material universe would contain only one perfect sphere, or less ideal, spheres symmetrically distributed on spherical surfaces - here it only being possible to think of symmetry about a line or a plane, not a unique point as with the single sphere. So one may equally question:

But what causes asymmetry?

In other words, why does the universe deviate from symmetry and yet remain recognisably symmetric?

Without entering into a meta-physical argument, one may vaguely suggest that this is due to the fundamental forces slipping out of phase (ie equilibrium) and the resultant force only possessing certain planes of symmetry.

At a higher level, one may argue that the zygo-pleurality found in living organisms maximises their chances of survival. For instance, a very contrived example might be a kangaroo with a pouch on its left or right hand side would be thrown off balance by the added uneven weight of its offspring and thus make it more prone to attack and thus lessen its chance of survival. A seemingly ridiculous example, but only ridiculous because man is so used to seeing symmetry in nature that all else seems ludicrous.

It is prudent to remark that many internal organs are not symmetrically placed. For example, the heart is to the left; the reason for this is difficult to understand, since the skeleton, the brain are both symmetric.

In fact, for the intestines, which were, according to the zoologist Wilhelm Ludwig, the primarily internal cause of asymmetry - due to the intestinal tube having to increase its surface (by folding up) out of proportion to the growth of the body and thus rearranging the other symmetrically placed organs - there is a 0.02% chance in every human that they will be oppositely arranged.

Thus there are many examples where the living and non-living world are in harmony - the bilateral symmetry of the kangaroo in equilibrium with the rotational symmetry of gravity at the surface of the Earth. It therefore appears that symmetry is a question of equilibrium and as systems move towards equilibrium and stability, so they move back towards the crystalline symmetry of space.

The most primitive forms of life do exhibit spherical symmetry (eg small pond creatures). However, for animals that are capable of self-motion, both the postero-anterior direction in which their bodies move and the direction of gravity are decisive factors in determining their form, as was illustrated earlier for the example of the kangaroo.

The other question to be answered is how does man detect this pervasive symmetry so easily? Is there some fundamental property in the visual cortex that allows for the recognition of a stimulus as reflexion symmetric, rotation symmetric and so forth, or does the neural system learn to do this operational transformation?

There is at present no way of deciding how the brain performs these mental feats of great generalisation for the degree of coding is so high. One can, of course, speculate that cells might exist that are similar to those discovered by Hubel and Wiesel for edge and movement detection to achieve these mental contractions.

In a later summary of their work, Hubel and Wiesel (1977) describe the functional architecture of the visual cortex of the macaque monkey.* They remark for the retinal ganglion cells found at the beginning of the visual processing system:

> "These cells have fields with circular symmetry, and respond roughly equally to all orientations of live stimulus."

Progressing through the visual cortex, they show that simple cells respond to optimally orientated lines at a particular position on the retina and further that complex cells are equally orientation specific but are more tolerant in position. Hyper complex cells are similar to complex, but are intolerant to extending the length of the line. However, they emphasise:

> "... how absurd it is to think of a simple or complex cell as a 'straight-line detector'. What is detected is the orientation of a short line segment. A straight line long enough to be seen as straight (or for that matter curved line) would have to be 'detected' by some cell receiving input from populations of area-17 cells (striate cortex) ... Whether such cells exist is not known." *

They imply that the deeper area-18 (which they identify with stereopsis) might be a site for such cells but have as yet not investigated this area of the cortex.

The phylogenetic determination of a creature's axis of symmetry by cellular mitosis is not simple. The cellular axis of symmetry can change as the organism develops, the original axis being a matter of either chance (position of the egg in the womb) or external forces (electrical or gravitational fields). (Weyl 1952)

It is interesting that in every human being there is a directional screw on a chemical level. Our bodies contain a dextro-rotary form of glucose and laevo-rotary form of fructose. This is illustrated by the disease phenylketonaria - which leads to insanity - and is caused by a man having been subjected to small doses of laevo-phenylalanine. The dextro-form has no ill-effects!

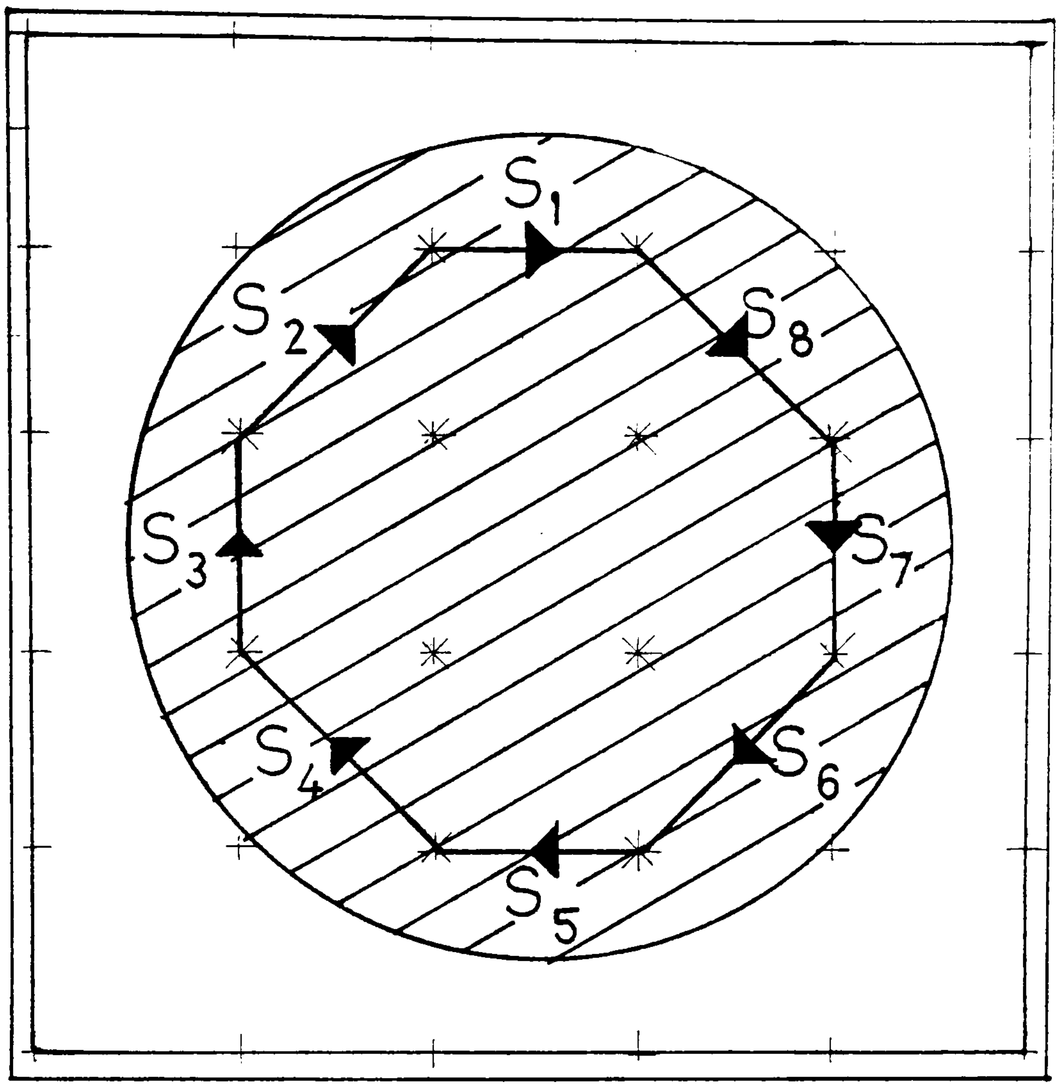
If symmetry recognition is a learned or programmed (a number of sequential tests) process, it is hard to see how this realisation

---

* Parenthetically it should be remarked that Hubel and Wiesel seem impressed by the neural symmetry of the visual cortex on a functional level. For when carrying out experiments to show ocular dominance, they state: "... as with zebras' stripes, certain general rules seem to be obeyed in all the monkeys". They go on to show that asymmetry in the use of cells is achieved if one eye is deprived from birth.

outer
contour

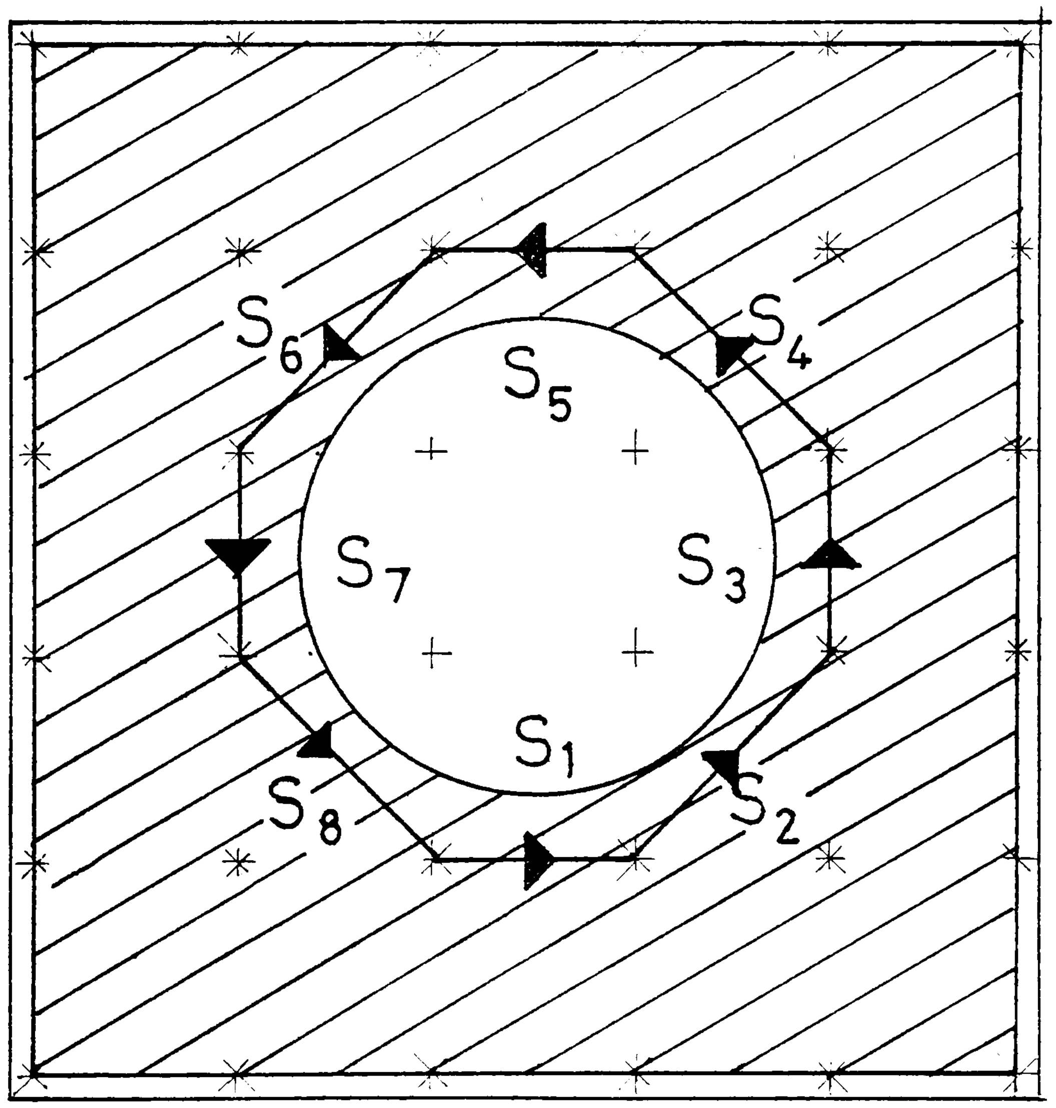
clockwise
chain



$+ \equiv 0$

inner
contour

counter
clockwise
chain



$* \equiv 1$

PATTERN-TO-SIGNAL CONVERSION FIGURE 1.4

could be proven. Later in the thesis, methods will be discussed to suggest how both possibilities can be synthesized.

1.2.b A Definition of Locally Symmetric Patterns

In the author's M.Phil. thesis (Hayes 1, 1976) - A Cybernetic Approach to Pattern Recognition - it was shown that a two dimensional binary matrix could be represented by a string or chain (see also Appendix 3) formed from a set of vector states $s_1, s_2, \ldots s_8$; where s_1 corresponds to the direction East, s_2 to the direction North-East and so on; typical conversions being those illustrated in Figure 1.4. Thus assuming that a serial to parallel conversion can be made where the states are collected in an orderly fashion (this has already been proven in the thesis just mentioned and is detailed in Appendix 3 and is further illustrated there), then a two dimensional pattern can be represented as a set of one dimensional state chains. It then follows that two dimensional symmetric shapes may be easily defined about vertical, horizontal and 45° diagonal axes, irrespective of the position of the patterns in the matrix, or other patterns that may be asymmetric. Also local edges of symmetry may be examined, that is, generated or recognised. All this is possible because edge information has been extracted in an orderly way and separate blocks of pattern isolated. This technique is very important in bandwidth reduction in picture coding and will be returned to: note that no information is lost.

1.2.b.i Single chain reflexional symmetries

Let the Qth chain V_Q be N states long, for example:

$$V_Q=(s_{Q1},s_{Q2}, \dots\dots\dots\dots s_{QN}) \tag{1.21}$$

It will first be assumed that N is even. Then V_Q is symmetric about M iff

$$s_{Q(M+R+1)}=s'_{Q(M-R)} \quad \text{for all R such that M, R, M+R and M-R are modulo } N^* \tag{1.22}$$

where $s'_{Q(M-R)}$ is a complementary state of $s_{Q(M-R)}$, for example s_4 is the complement s_2 for vertical symmetry and vice versa, the complete complement tables are given in Figure 1.5. This relates precisely to the one dimensional line symmetry just described in section 1.1, except that the complementary state must be taken. Local evenly composed symmetry may be imposed by restricting the modulo range of R (eg R<K, where K<N and is the extent of local symmetry).

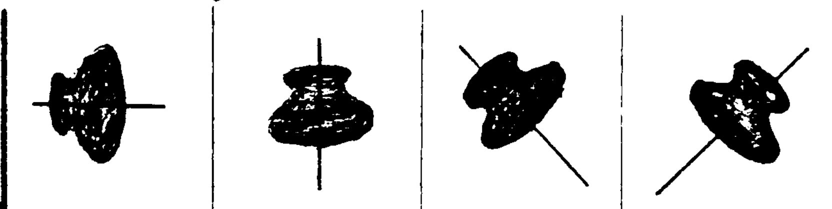
An example of this form of evenly composed planar line symmetry is shown in Figure 1.6.

For the case where N is odd. Then V_Q is symmetric about M iff

$$S_{Q(M+R)}=s'_{Q(M-R)} \quad \text{for all R such that M, R, M+R and M-R are modulo } N^* \tag{1.23}$$

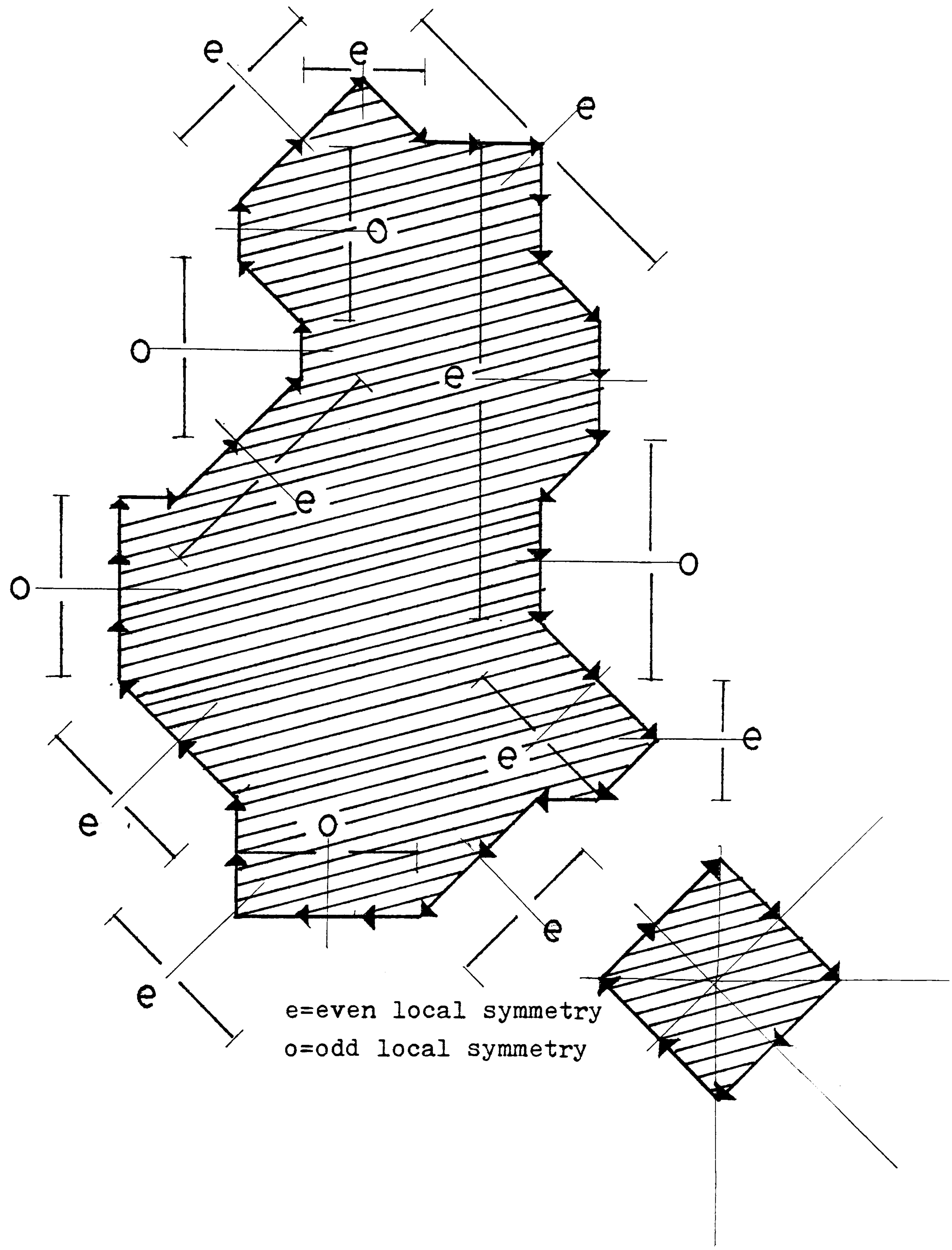
Locally oddly composed symmetry for a single chain is defined as for the even case. An example is shown in Figure 1.6.

* All suffic arithmetic in this section is now assumed modulo N



state of the input \ type of symmetry				
1 →	1 →	5 ←	7 ↓	3 ↑
2 ↗	8 ↘	4 ↖	6 ↙	2 ↗
3 ↑	7 ↓	3 ↑	5 ←	1 →
4 ↖	6 ↙	2 ↗	4 ↖	8 ↘
5 ←	5 ←	1 →	3 ↑	7 ↓
6 ↙	4 ↖	8 ↘	2 ↗	6 ↙
7 ↓	3 ↑	7 ↓	1 →	5 ←
8 ↘	2 ↗	6 ↙	8 ↘	4 ↖

COMPLEMENTARY TRANSITION TABLE FIGURE 1.5



TWO DIMENSIONAL SYMMETRIES FIGURE 1.6

1.2.b.ii Other forms of symmetry for a binary pattern

For other angles of axes of symmetry, a more complicated approach is required. By testing the adjacent states of V_Q for straightness as is described in the writer's M.Phil.thesis and Appendix 3 it is possible to produce a contracted vector chain V'_Q:

$$V'_Q=(v_{Q1},\ v_{Q2},\ \dots\dots\dots\dots\ v_{QN'}) \qquad (1.24)$$

where v_{QR} is a two dimensional vector with integer components and $N'<N$. An example of this form of representation is given in Figure 5b of Appendix 3.

Now in order to test for reflexion symmetry, a reflexion matrix operator "R'_ϕ" can be applied to V'_Q such that:

$$v_{Q,M-R}=R'_\phi(v_{Q,M+R+B}) \text{ for all } R\leqslant N/2+B \qquad (1.26)$$

should be satisfied over any region of reflexional symmetry, where

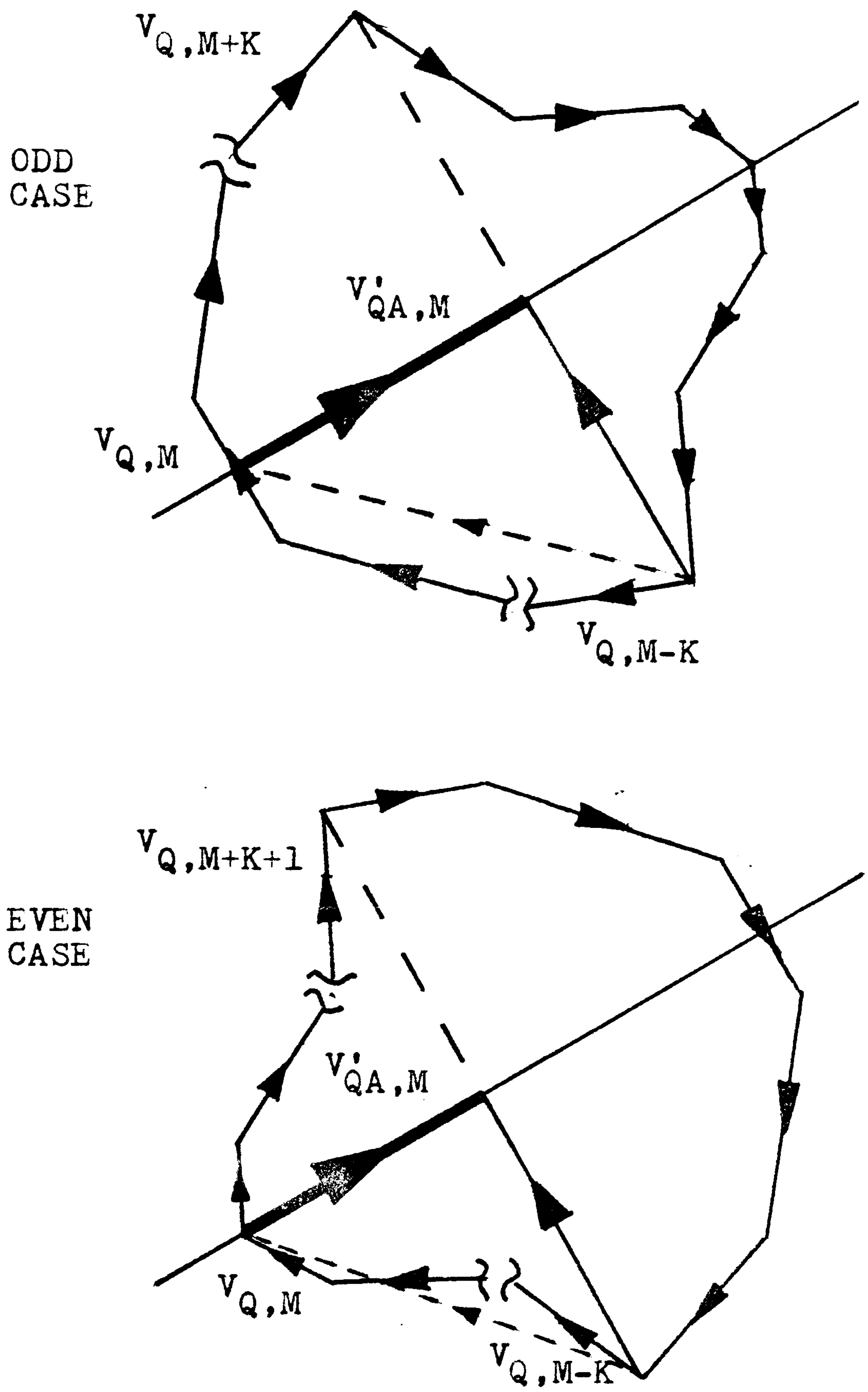
$$R'_\phi=\begin{bmatrix}\cos\phi & -\sin\phi\\ -\sin\phi & -\cos\phi\end{bmatrix} \qquad (1.26a)$$

and 'B' is a boolean variable depending on whether the number of elements n' in the chain is odd (ie B=1) or even (ie B=0).

The value of "ϕ" may be determined by first determining a possible axis of symmetry "v'_{QA}" by, for example, the formula:

$$v'_{QA}=\tfrac{1}{2}\sum_{R=M-K}^{R=M+K+1-B} v_{QR}-\left(\sum_{R=M-K}^{R=M} v_{QR}-\tfrac{1}{2}B.v_{QM}\right) \qquad (1.27)$$

and then:



VECTOR REPRESENTATIONS FIGURE 1.7

$$\phi=\arctan((V'_{QA})y/(v'_{QA})x) \qquad (1.28)$$

where $(v)_X$ and $(v)_Y$ are the X and Y components of the enclosed vector "v".

Figure 1.7 demonstrates these considerations graphically.

1.2.b.iii Single chain rotational symmetries

Rotational symmetry is a self property defined by:

$$V'_Q=R_\phi(\Theta_M(V'_Q)) \qquad (1.29)$$

where operator "Θ_M" performs the Mth cyclic interchange of the vectors of the chain "V'_Q" and

$$R_\phi= \begin{bmatrix} \cos\phi & -\sin\phi \\ \sin\phi & \cos\phi \end{bmatrix} \qquad (1.29a)$$

is the rotation matrix.

Thus to determine whether a vector chain possesses rotational symmetry, all pairs of equal length vectors must be examined in order to establish possible ϕ's through the relationship

$$\phi=\angle v_I, v_J \qquad (1.30)$$

where v_I and v_J belong to V'_Q and are of equal length - "$\angle$" is an angle operator defined fully in Appendix 3. Expression 1.29 can then be applied as a test, where "M" is "J" minus "I". If it is found to be true, then a rotational symmetry has been identified and for total rotational symmetry will repeat $360^\circ/\phi$ times.

However, if expression 1.29 is only partially satisfied over, say, over K elements, then a partial rotational symmetry may be defined to exist.

It should therefore be stated that for total symmetry, ϕ will be an integer division of 360 and expression 1.29 may be tested directly for such, also reflexional symmetries may well exist about similar points as rotation symmetries* and this may prove a good starting point for any tests (having already established reflexional symmetries).

1.2.b.iv Multiple chain symmetries

The problem of defining and establishing symmetries when multiple chains are involved revolves around comparing vector chains. For example, to establish a reflection, a chain V_Q should relate to another chain $V_{Q'}$, such that:

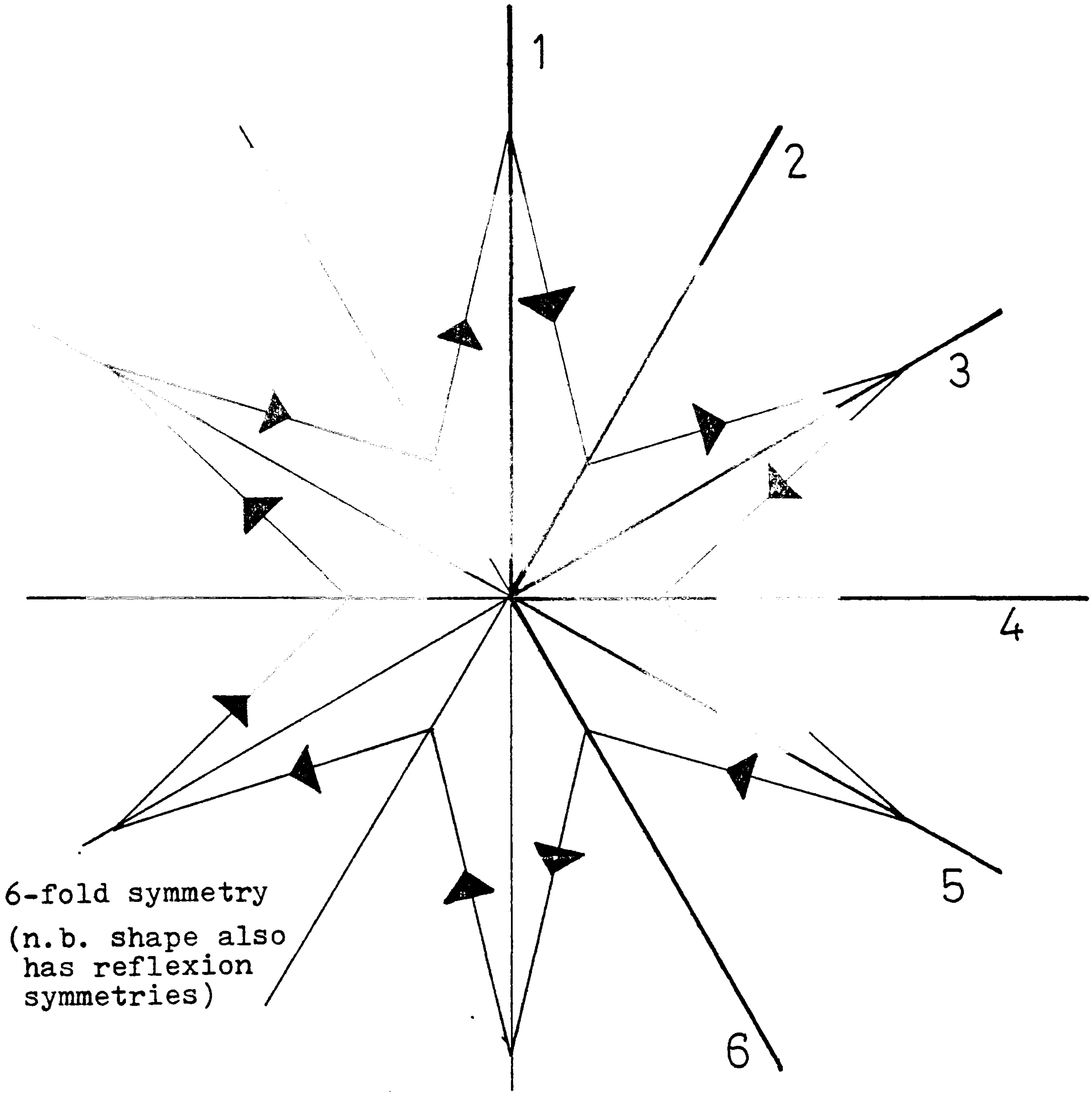
$$V_{Q'}=V'_Q \tag{1.31}$$

where the vectors in $V_{Q'}$ are identical to the vectors in V_Q having been reflected in a vector axis $v_A(Q,Q')$, formed by a composition:

$$v_A(Q,Q')=v_{QR}+v_{Q'R'} \text{ for all modulo R and R'} \tag{1.32}$$

v_A remaining fixed when a true vector axis is found for all cyclic R or R'.

* When this occurs, the object belongs to a dihedral group.



ROTATIONAL SYMMETRIES

FIGURE 1.8

There are many possible multiple symmetries, and these were first studied by Arab mathematicians and displayed in their art. Today their work is continued by crystallographers who study the molecular structure of matter.

It should be noted that certain types of multiple chain symmetry are tricky for the human eye to detect and rest highly on subtle predication processes that are beyond the scope of this text. Some interesting examples are shown in Figure 1.9.

1.2.c Multi-state Patterns - their Connectivities and Symmetries

The multi-state two dimensional pattern can be treated in the way described in the previous sections by setting thresholds that will binarise the pattern as is shown in Figure 1 of Appendix 3. This then imposes a certain state connectivity on the matrix which allows borders of equipotential to be detected.

The actual setting of the threshold may be either based on a global or a local average. The global average is not in itself directly edge sensitive and is therefore better suited to logical differencing methods described by the author in his previous works (Hayes, 1,1976)

It is worthy of remark that a one-dimensional local symmetry is being enforced on the multi-state pattern matrix by this binarisation operation, that selectively filters connectivity.

That is to say that there exists a symmetric topological invariance for all multi-state patterns such that for every one dimensional cut, there is at least a pair of upward and downward going gradients, moving along the cut in a fixed direction. Without this invariance, sight would be impossible.

In a more practical vein, it should be recognised that a gradual stepped increase in threshold will give rise to a set of contours that may possess the multi-chain symmetries mentioned in the previous section. To cope with these notions of concentricity must be developed - this is a special study and is developed in Appendix 3.

Before closing this chapter, it should be summarily stated that in 2-D space, central symmetries may be broadly classified by their belonging to certain groups:

$$C_1 C_2 C_3 \ldots \quad ; \quad D_1 D_2 D_3 \ldots \tag{1.33}$$

C_1 means no symmetry at all, D_1 bilateral symmetry and so on. In general C_n is a cyclic group consisting of the repetitions of a single proper rotation by an aliquot part $\alpha = 360^{\circ}/n$ of 360° and D_n is a dihedral group of these rotations combined with the reflexions in n axis forming angles of $\frac{1}{2}\alpha$ (Weyl 1952). So that for two members of a particular D_n, that is d_{ns} and d_{nt} there will always exist a relationship

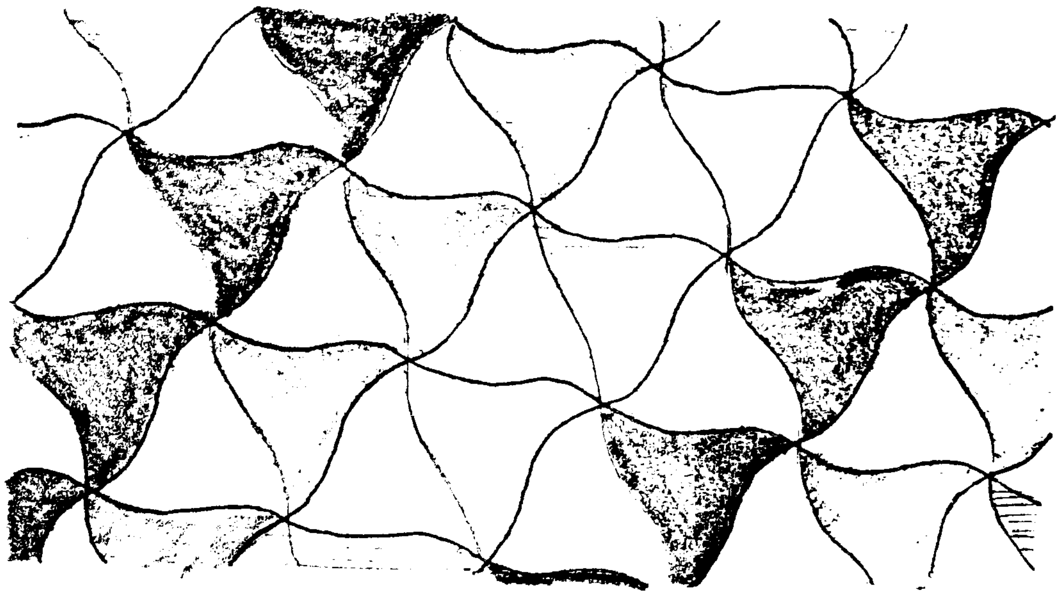
$$d_{ns} = r(d_{nt}) \tag{1.34}$$

where r is a rotation or a reflexion operator.

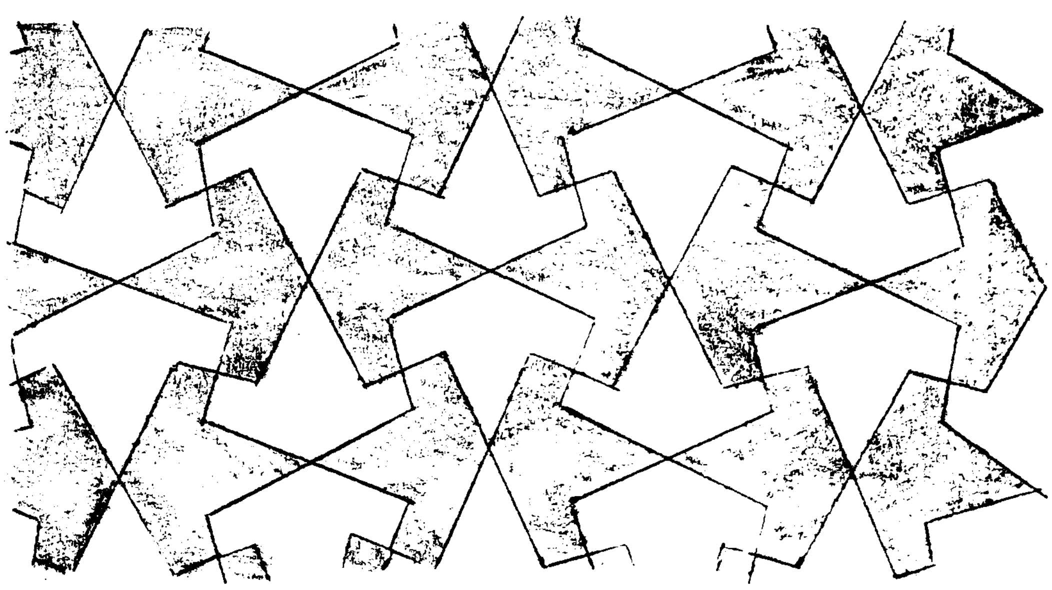
Two physical facts are worthy of final remark. The first is that in 3D space there are only five polyhedra possible, often called the Platonic solids. They are the tetrahedron, the cube, the octahedron, the pentagondodecahedron (12 regular pentagonal sides) and the icosahedron (20 regular triangular sides). Now, by considering rotations which carry a body into itself, one can extract a further 3 groups, so that

$$\begin{array}{ll} C_n & (n = 1, 2, 3 \ldots) \\ D'_n & (n = 2, 3 \ldots) \\ T, W, P & \end{array} \tag{1.35}$$

where C_n has already been defined in 2D space, D'_n is now interpreted as a rotation outside the plane to give the effect of reflexion within the plane, T is the group of 12 operations that take a tetrahedron into itself, W is the group of 24 operations that take a cube (or octahedron) into itself and P is the group of 60 operations that take a pentagondodecahedron (or icosahedron) into itself. The list can be increased if improper rotations are defined and this is pursued by Weyl (1952) into the realms of crystallography. Where the second physical fact emerges that inorganic pentagonal symmetry is unknown, no other rotational symmetries are possible than those of order 2, 3, 4 and 6, this is solely because of the restrictions imposed by a lattice. Thus a geranium flower exists solely because of its spatial freedom.



Arabic Design illustrating only Rotation Symmetries



Arabic Design with Rotation and Reflexion Symmetries

MULTIPLE CHAIN SYMMETRIES   FIGURE 1.9

CHAPTER TWO

THE RECOGNITION OF A SYMMETRIC MESSAGE

Foreword

The problems involved in identifying a symmetric message are introduced by way of the concept of an error signal, termed a symmetry metric, that is based on the definitions of symmetry established in Chapter One.

This leads to a discussion of decision making techniques, which can be applied to the associated symmetry metrics for signal processing and pattern recognition.

2.1 Relations between Signal Processing and Pattern Recognition

> "In communication theory, we regard the receipt of noisy signals as providing evidence of the messages selected at the transmitter, such evidence converting the receiver's hypotheses concerning the possible messages from a prior set to a posterior set, from which the receiver can make some "best" guess, with a chance of error."
>
> Colin Cherry

The methods involved in detecting a message in noise are well documented in the literature of signal processing* - the principal technique being that of the "matched filter" - and to a lesser extent in pattern recognition,** where the most commonly used techniques are "the nearest neighbour" and "mask matching". Both these subjects rest heavily on the concept of linearly weighted summation, where the weights in the case of a matched filter are complex conjugates of the signal sequence and form a prototype vector for the inner product (as has been illustrated in Figure 1.3), while, in the case of pattern recognition, the weights are so adjusted that the associated hyperplanes separate the maximum number of patterns belonging to different classes, (eg signal = a class, noise = another class) - these weights either being arrived at statistically or geometrically. Thus the essential differences between signal processing and pattern recognition are:

1 Signal processing places a greater emphasis on the nature of the noise (which is usually assumed to be Gaussian) than does (or can) pattern recognition.

2 Pattern recognition is concerned more with classes of patterns than with particular signals that may only vary in amplitude.

* See Brown and Glazier 1964
** See Ullmann 1973

From these distinctions there follow two other differences concerning the way the techniques are employed:

3 The methods of pattern recognition tend to be used on properties, attributes and features of the patterns to be classified, whereas in signal processing the methods are applied directly to the signal, there being a minimum of preprocessing.

4 The types of techniques used in pattern recognition tend to be more ambitious, intuitive and ingenious than the conservative and theoretically justifiable methods of signal processing and may rely on such concepts as learning and training.

Plainly, in discussing symmetric signal detection it will be useful to employ the rigour of signal processing while attempting to retain the ingenuity of pattern recognition.

The first concept to be introduced unquestionably involves both signal processing and pattern recognition, for it concerns itself with the class of symmetric signals and patterns defined in Chapter One.

2.2 A Measure in the Deviation from Symmetry of a Signal

If one examines the definitions of even continuous symmetry, then it is possible to define a deviation from or error in symmetry $S(x)$ about a point x, as follows:

$$S(x)=\int_{s=0}^{s=h} \left| f(x+s)-f(x-s) \right| ds \tag{2.1}$$

where h is the extent of symmetry. This integral, in essence performs the operation of folding the function $f(x)$ over on itself and accumulating the absolute differences. Notice that the definition corresponds closely to conventional mask matching found in pattern recognition while having the self-property of self comparison found in the auto-correlation function. It should of course be realised that the loss of sign achieved by taking the modulus could be achieved by squaring or raising to any even power the enclosed difference. The advantage in taking a squared error would be that the integral would then be analytic; however, in most cases the function will probably be sampled and the integral a summation and, under these circumstances, the absolute error is much easier to compute. In fact, if this is the case, then it is possible to calculate $S(x)$ without undue delay in a micro-processor which will not require the added complexity of a multiplier or squarer. It may be desired to weight the error over the range of expected symmetry and this can be accomplished by introducing into the integral a multiplicative weighting function that must itself be evenly symmetric over the window $2h$.

More formal definitions of the symmetry metrics will now be given.

2.2.a An even, continuous symmetry metric

A continuous function f(x), can be tested for the property defined in section 1.1.b.i, by considering an absolute local error '$\mathcal{E}_e$' in local symmetry, a distance 's' from a point 'x', that is:

$$\mathcal{E}_e(x,s)=\left|f(x+s)-f(x-s)\right| \tag{2.2}$$

and then integrating with respect to 's' to give the total error about 'x' over a range '2h', one arrives at:

$$\int_0^h \mathcal{E}_e(x,s)ds=\int_0^h \left|f(x+s)-f(x-s)\right| ds \tag{2.3}$$

which is a function of the extent of the metric 'h' and the position 'x', and can be normalised (or averaged) over the range '2h' to become "an even symmetry metric" - $s_e(x)$, where:

$$s_e(x)=1/2h \int_0^h \left|f(x+s)-f(x-s)\right| ds \tag{2.4}$$

It will be noticed that $s_e(x)>0$ for any asymmetric function for all x and $s_e(x)=0$ for any even, locally symmetric function when x is the centre of symmetry and $h \leqslant k$.

In analytic - rather than computational - work, it may be preferred to define:

$$s_e'(x)=1/2h \int_0^h (f(x+s)-f(x-s))^2 ds \tag{2.5}$$

for example, it will then be seen:

$$s_e(x)={}^1/_{2h}\int_0^h (f(x+s))^2ds + \int_0^h (f(x-s))^2ds - 2\int_0^h f(x+s)f(x-s)ds$$

$$={}^1/_{2h}\left(\int_{-h}^{h} (f(x+s))^2ds- 2\int_0^h f(x+s)f(x-s)ds\right) \tag{2.6}$$

if f(x-s)=g(x+s):

$$={}^1/_{2h}\left(\int_{x-h}^{x+h} (f(s))^2ds - 2\int_{x-h}^{x+h} g(s)f(s)ds\right) \tag{2.7}$$

$$= \phi_f(x,0) - \phi_{gf}(x,0) \tag{2.8}$$

where $\phi_f(x,0)$ is a bounded auto-correlation function about 'x' with no increment and $\phi_{gf}(x,0)$ is a bounded "cross-correlation" also with no increment, but since g is a reflection of f about 'x' ϕ_{gf} is better viewed as a modified auto-correlation.

2.2.b An even, discrete symmetry metric

A discrete function F(X), can be tested for the property defined in section 1.1.b.ii by considering a set of absolute local errors in even symmetry about an integer 'X', extending over an extent 'H' and averaging these errors to give:

$${}_eS_e(X)={}^1/_{2H}\sum_{S=0}^{S=H} \left|F(X-S)-F(X+S+1)\right| \tag{2.9}$$

which may alternatively be defined by a mean square local error:

$${}_eS_e(X)={}^1/_{2H}\sum_{S=0}^{S=H} (F(X-S)-F(X+S+1))^2 \tag{2.10}$$

that as with the continuous case outlined in section 2.2.a may be manipulated to produce:

$$ {}_{e}S_{e}(X) = \Phi_F(X,0) - \Phi_{GF}(X,1) \tag{2.11} $$

where G(X+S)=F(X-S) and

$$ \Phi_F(X,0) = {}^{1}/_{(2(H+1))} \sum_{X-H}^{X+H+1} (F(S))^2 \tag{2.12a} $$

and

$$ \Phi_{GF}(X,1) = {}^{1}/_{(H+1)} \sum_{X}^{X+H+1} F(S+1)G(S) \tag{2.12b} $$

A discrete function F(X) may be tested for the property defined in section 1.1.b.ii by adopting the same technique used in section 2.2.a to produce a symmetry metric:

$$ {}_{o}S_{e}(X) = {}^{1}/_{2H} \sum_{S=1}^{S=H} \left| F(X-S) - F(X+S) \right| \tag{2.13} $$

or alternatively:

$$ {}_{o}S_{e}(X) = {}^{1}/_{2H} \sum_{S=1}^{S=H} (F(X-S) - F(X+S))^2 \tag{2.14} $$

which may be rewritten as:

$$ {}_{o}S_{e}(X) = \Phi_F(X,0) - \Phi_{GF}(X,0) \tag{2.15} $$

where Φ_F and Φ_{GF} are only averaged over 2H and H respectively.

A regularly sampled continuous function F(X), may be tested for even symmetry of no preferred composition by defining:

$$\mathcal{S}_e(X) = \min\left\{ {}_eS_e(X), {}_oS_e(X) \right\} \tag{2.16}$$

or if the mean squared form is preferred:

$$\mathcal{S}_e'(X) = \min\left\{ {}_eS_e'(X), {}_oS_e'(X) \right\} \tag{2.17}$$

where $\mathcal{S}_e$ and $\mathcal{S}_e'$ are approximations to their continuous counterparts s and s'.

2.2.c Some observations on global and odd symmetry metrics

A continuous function f(x), can be tested for the property defined in section 1.1.b.iii by considering an absolute local error '$\mathcal{E}_o$' in odd symmetry, a distance 's' from 'x', to give:

$$\mathcal{E}_o(x,s) = \left| 2f(x) - f(x+s) - f(x-s) \right| \tag{2.18}$$

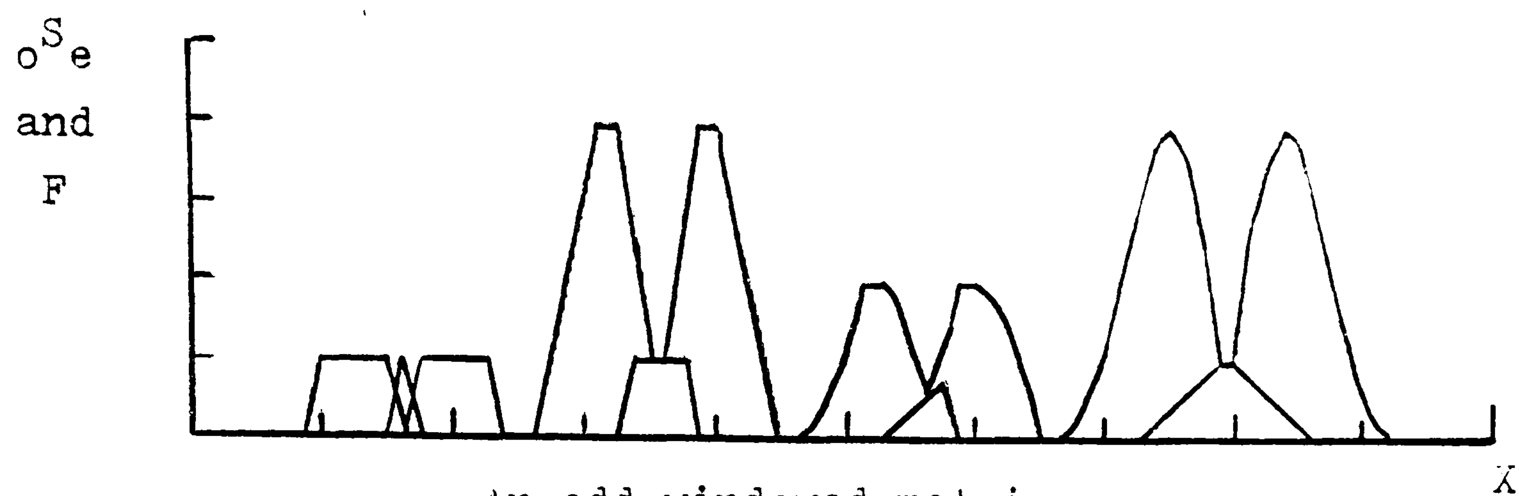
and then accumulating, as in section 2.2.a to arrive at:

$$s_o(x) = {}^1/_{2h} \int_0^h \left| 2f(x) - f(x+s) - f(x-s) \right| ds \tag{2.19}$$

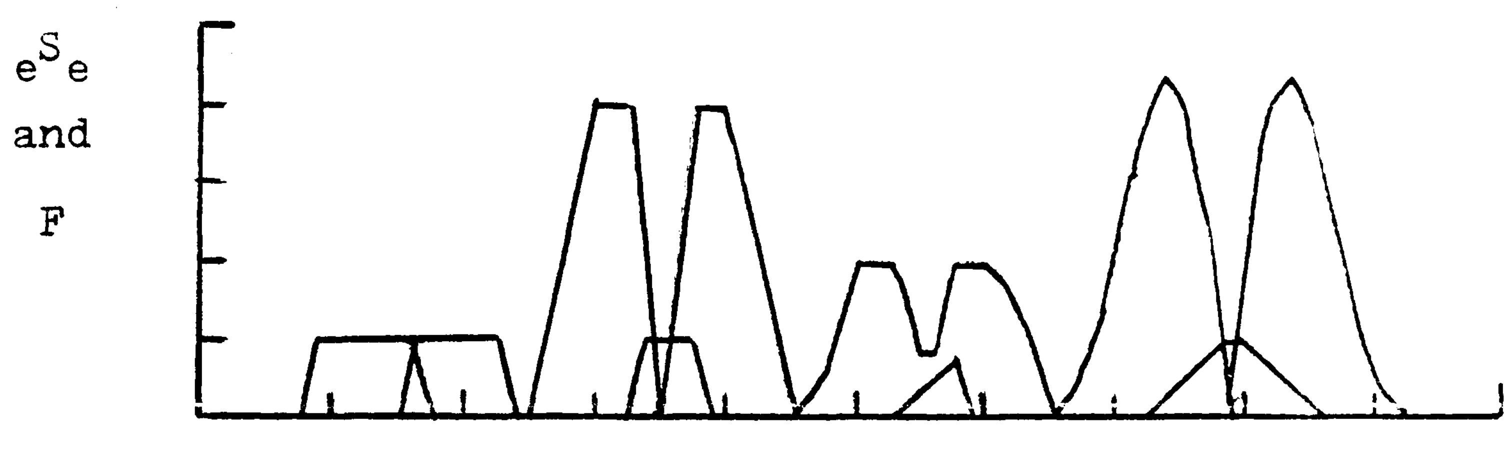
where s_o is the odd, continuous, symmetry metric. Alternatively, one may prefer to use:

$$s_o'(x) = {}^1/_{2h} \int_0^h \left(2f(x) - f(x+s) - f(x-s) \right)^2 ds \tag{2.20}$$

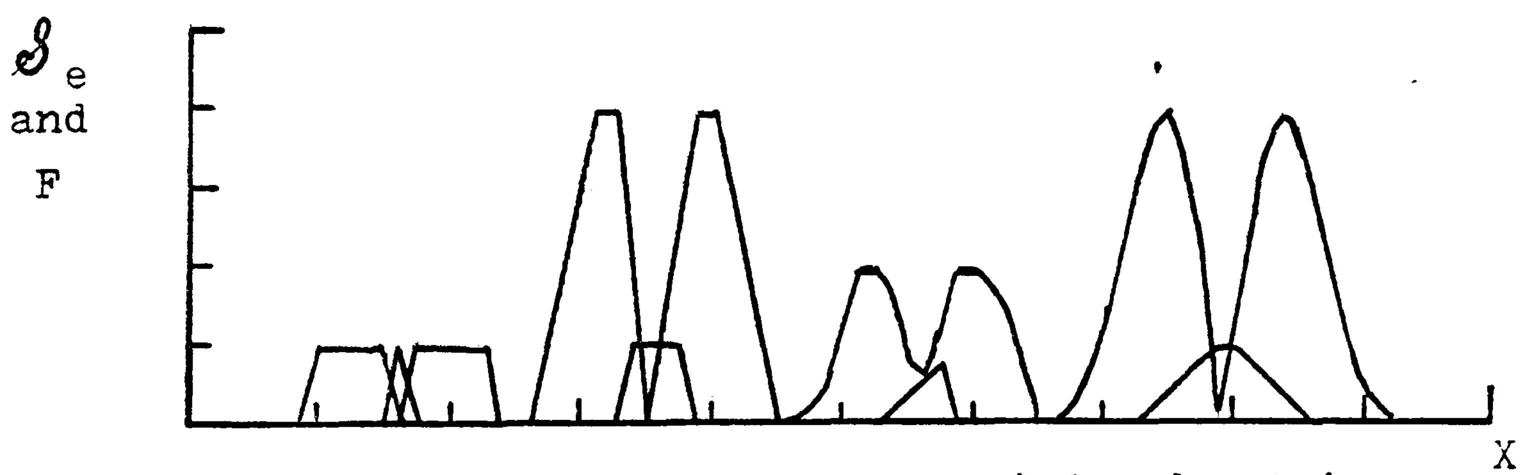
A discrete function F(X), can be tested for the property defined in section 1.1.b.iii by considering a set of absolute local errors in odd symmetry about an integer 'X', extending over an extent 'H' and averaging these errors to give:



An odd windowed metric

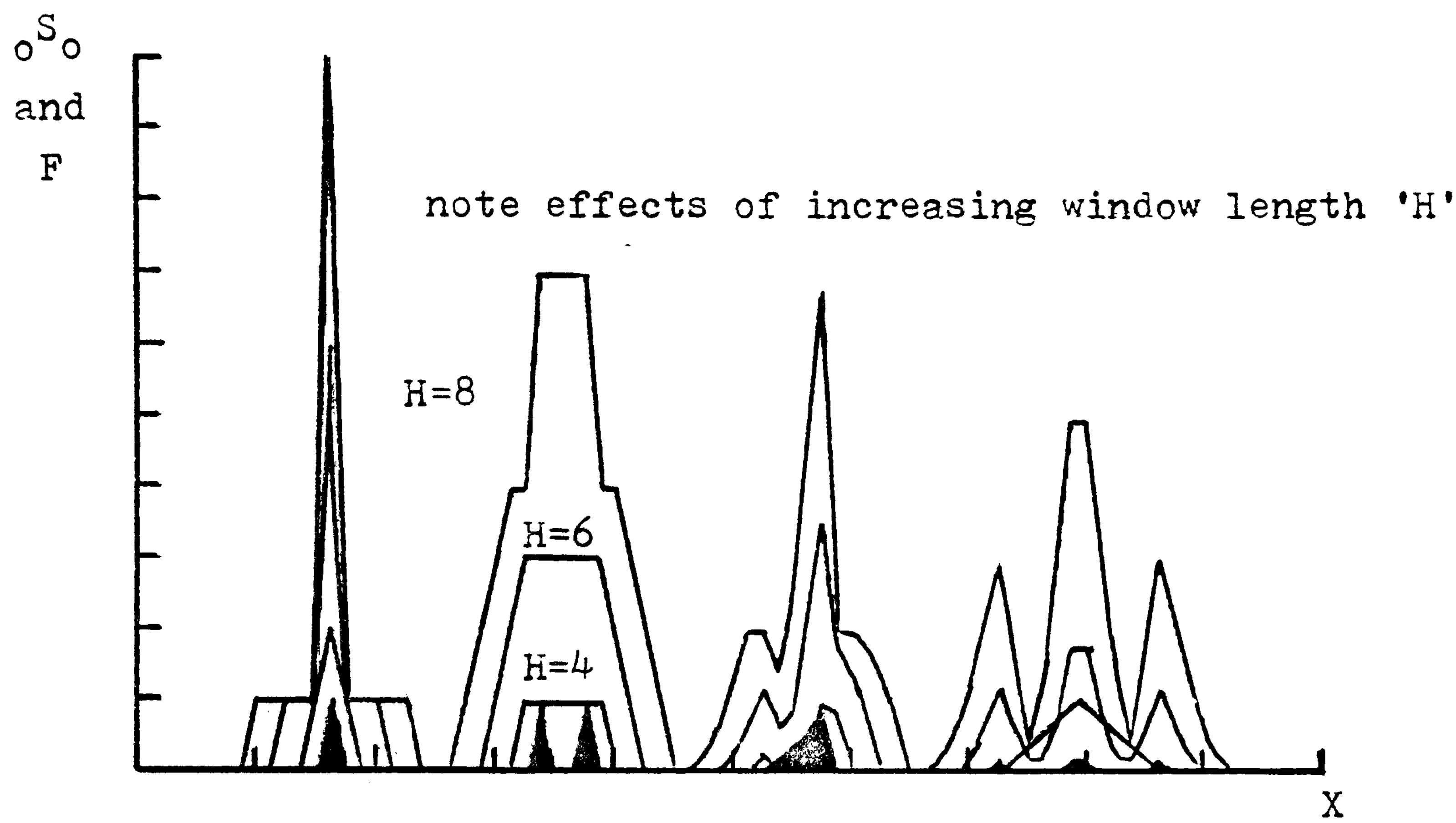


An even windowed metric

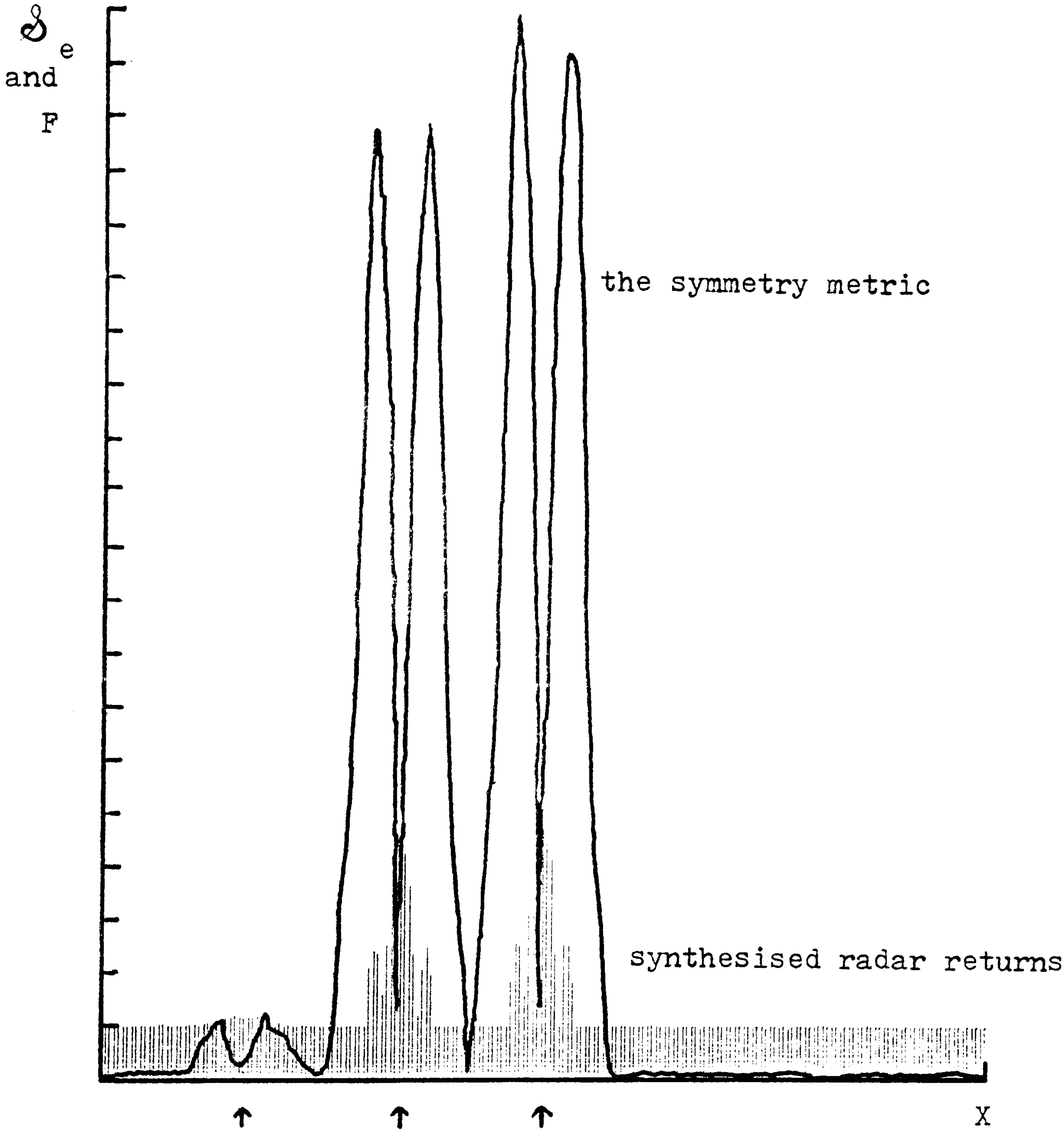


A minimum odd-even windowed metric

EVEN SYMMETRY METRICS FIGURE 2.1



ODD SYMMETRY METRICS FIGURE 2.2



THE SYMMETRY METRIC FIGURE 2.3

$$ {}_{o}S_{e}(X)=1/_{2H}\sum_{S=1}^{S=H} \left|F(X-S)+F(X+S+1)-F(X)-F(X+1)\right| \tag{2.21} $$

which may alternatively be defined by a mean square local error:

$$ {}_{o}S'_{e}(X)=1/_{2H}\sum_{S=1}^{S=H} (F(X-S)+F(X+S+1)-F(X)-F(X+1))^{2} \tag{2.22} $$

and for the evenly composed case:

$$ {}_{o}S_{o}(X)=1/_{2H+1}\sum_{S=1}^{S=H} \left|2F(X)-F(X-S)-F(X+S)\right| \tag{2.23} $$

The application of expressions 2.9, 2.13 and 2.16 to various simple inputs is shown in Figure 2.1. These may be contrasted with that shown in Figure 2.2 where expression 2.23 has been applied. Notice that whereas the even metrics tend to be minimal at a point of local symmetry, the odd symmetry metrics are maximal.

Global symmetry is easily defined for the continuous case by allowing the extent of expression 2.4 to tend to infinity - a similar limiting process may be defined for the discrete case.

2.3 A Measure in the Deviation from Symmetry of a Pattern

> "Your face is the same as everybody has - the two eyes so ' ... (marking their places in the air with his thumb) nose in the middle, mouth under. It's always the same. Now if you had two eyes on the same side of the nose for instance - or the mouth at the top - that would be some help."
>
> Humpty Dumpty

In section 1.2.b a 2-D symmetric pattern is defined for a composition of both an odd and an even numbered vector chain. Now, clearly, in a real universe very few natural phenomena, or even artifacts, are going to meet this rigorous definition. It is therefore necessary, as with the 1-D signal, to define an error in symmetry - a deviation from perfection. This would of course be necessary even if perfect symmetry were to exist, for all this theory rests upon a signal or pattern being sampled, and this inevitably introduces sampling or quantising noise, thus destroying the possibility of recognising perfect symmetry as something other than an abstract concept.

2.3.a A measure of single chain symmetry

A measure of reflexional symmetry for a 2-D vector may be defined by:

$$Z'(M)=1/N \sum_{R=0}^{R=N'} \left| V(M-R)-R_{\phi}(V(M+R+B) \right| \qquad (2.24)$$

where $Z'(M)$ is the deviation in reflexional symmetry about a reference point M and M and R' are modulo N' (the length of the chain), also B is a Boolean variable such at B=1 if N' is odd and B=0 if N' is even. Note that R'_ϕ is the reflexion matrix:

$$R'_\phi = \begin{bmatrix} \cos\phi & -\sin\phi \\ -\sin\phi & -\cos\phi \end{bmatrix} \tag{2.24a}$$

and that ϕ is a free variable to be determined by tests for consistency (which produces an axis of reflection symmetry at $\phi/2$ to the x-axis).

For the case of rotational symmetry:

$$Z(M)=1/N' \sum_{R=0}^{R=N'} \left| V(R) \quad -R_\phi(V(M+R+B) \right| \tag{2.25}$$

where Z(M) is the deviation in rotational symmetry about a reference point M and R, B and N' are as above, and

$$R_\phi = \begin{bmatrix} \cos\phi & -\sin\phi \\ \sin\phi & \cos\phi \end{bmatrix} \tag{2.25a}$$

In testing for both reflexional and rotational symmetries the range N may be restricted to E if only local symmetry is sought. Also a squared rather than an absolute value of the argument of the summation may be taken if more convenient.

2.3.b A measure of multiple chain symmetry

Two sorts of multiple chain symmetry are considered formally in 5c iv of Appendix Three* and the author has little to add, except

* NB A formal metric for multiple chain symmetry is not actually given, but this can be readily obtained by generating error functions, in the same manner as in the previous section, from equations 53 and 57. Certain difficulties do arise in the number of possible combinations capable of generating symmetry that appear to give rise to intractable amounts of computation and for this reason there is a need to study more tenable methods.

that if each vector chain is first identified as a separate and distinct pattern or element, then it may be possible, as an alternative approach, to use a higher level reasoning to check for multiple chain symmetries that can of course occur in a wide number of ways.

This diversity is in itself interesting for in complicated art works such as those by Escher it is possible to miss the more subtle n-fold symmetries. How strange that art appreciation becomes a problem-solving exercise of recognising symmetry and symmetric relationships.

Can Cybernetics give the artist, as well as the scientist, insight into his appreciation of space and its composition - are there symmetries that we collectively do not see?

2.4 The Detection of Symmetric Signals

As a measure of symmetry has now been defined, it is possible to discuss some detection processes. The classical technique is to set a threshold for the error signal and when this threshold is crossed to define the presence of a symmetric signal.

However, since noise in most cases is many decibels below signal and to the eye and the symmetry metric flat, it too will produce a zero error signal as is shown in Figure 2.3. It is therefore necessary to search for the sharp downward pointing cusp displayed also in Figure 2.3.

By performing a series of tests on the discrete symmetry metric of the form:

$$D(R,M) = \left\lceil S(R-M) > S(R-M+1) + \rho_M \right\rceil \wedge \left\lceil S(R+M) > S(R+M-1) + \rho_M \right\rceil \tag{2.26}$$

where ρ_M is the mth threshold and the operational brackets "$\lceil\ \rceil$" assign the enclosed expression a value '0' or '1' according to whether the expression is true (=1) or false (=0). A final joint decision as to the symmetric signal's central axis at R may be expressed as:

$$\blacktriangle(R) = \left\lceil \sum_{M=1}^{M=N} D(R,M) \geqslant \rho \right\rceil \tag{2.27}$$

where ρ is a threshold such that $0 < \rho \leqslant N$; then a unanimous decision will be reached if $\rho = N$ and a majority decision if $\rho = N/2+1$ in evaluating $\blacktriangle(R)$. It should of course be mentioned that there are many possible sets of threshold discriminants that would perform this role. The major problem is not, however, the determining of the members of these sets, but deciding of the optimum thresholds. This is by far the most difficult problem in signal processing and the whole of chapter 4 is to be devoted to the discussion of this task.

Before continuing with the problem of symmetric pattern detection it is necessary to point out that the conventional theory of matched filtering still holds good and is the optimum method if enough is known about the signal (ie its exact shape), but it is however several times harder to execute for it involves the application of a transversal filter which in turn involves multiplication. Nevertheless, a very good test for the axis of symmetry of a known signal is to apply matched filter theory and also apply the symmetry metric (this is almost equivalent to the sum and difference channels in a conventional tracking radar).

2.5 The Recognition of Symmetric Patterns

Unlike the symmetric signal detection discriminant just qualified, the discriminant for recognising a symmetric pattern must in general be more specific; for the number of two dimensional patterns is immense, often a pattern has more than one global axis of symmetry and many local axes (see Figure 2.4). However, there are similarities between the identification of symmetric signals and patterns, the major one being that a threshold must be applied and the optimum extent of symmetry determined for a particular facet of symmetry for a given region.

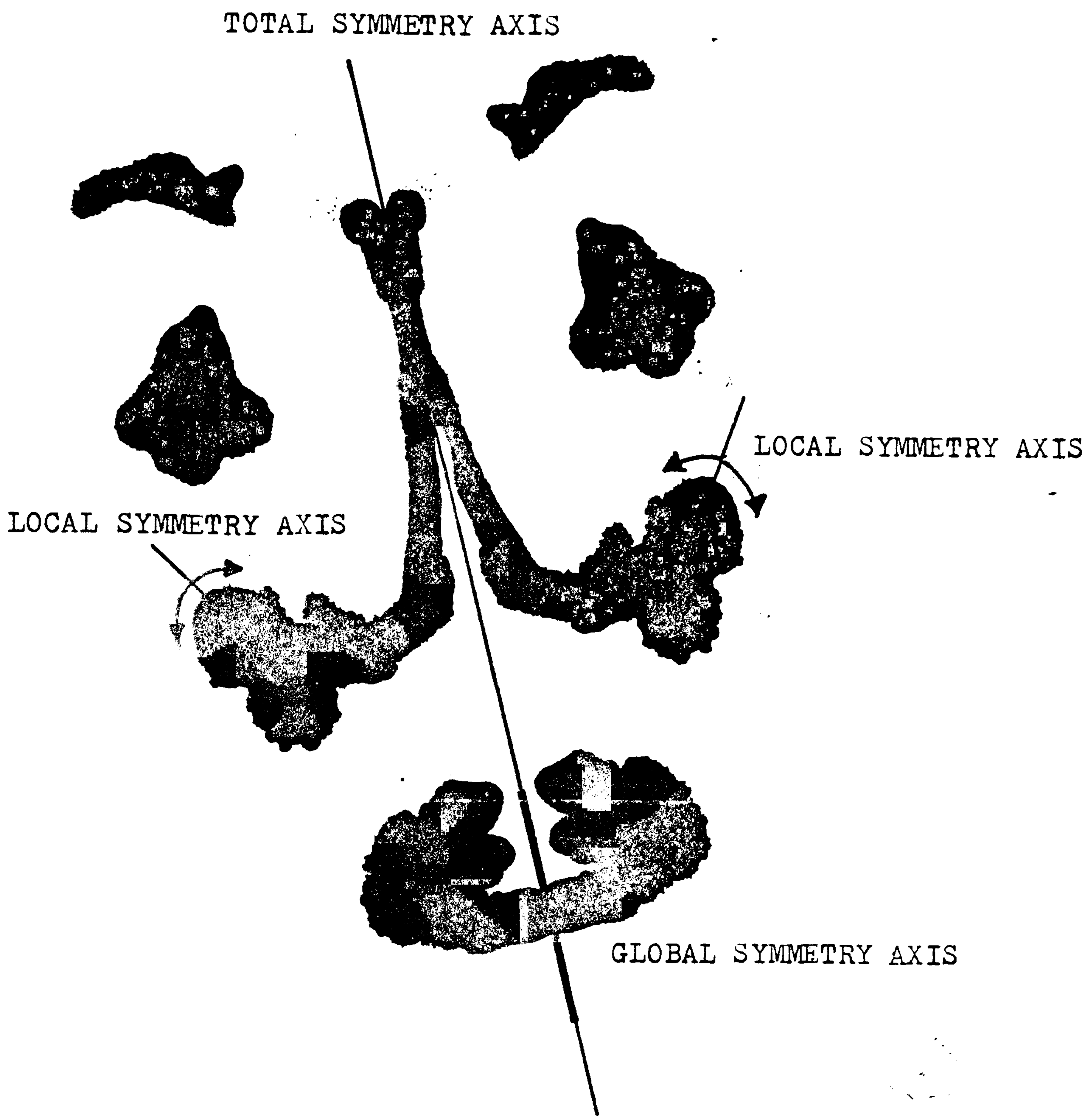
Clearly a simpler discriminant can be applied to Z or Z' - the symmetric pattern metric, because there is no chance of this being close to zero unless there is perfect symmetry, for most of the noise has been suppressed by the pre-recognition processing. Thus an absolute value of threshold will apply. For example:

$$\Delta(M) = \lceil Z(M,E) < \Theta \rceil \qquad (2.29)$$

where Δ is a function of the input matrix's dimension.

It may be useful to apply pseudo-symmetric tests for symmetry by designing feedback procedures that sub-divide the vector chain at points of local symmetry, the extent of local symmetry being gradually allowed to increase until the threshold is exceeded - this minimising the number of sequential tests.

The essential point is that at every point of local symmetry, there exists a useful feature to help recognise the pattern which may or may not be totally symmetric.



AXES OF SYMMETRY IN PATTERNS FIGURE 2.4

(n.b. This is a Rorschach ink blot)

CHAPTER THREE

A CYBERNETICS OF THE RECOGNITION PROCESSES

Foreword

In the following chapter the concept of a Cybernetic system as a system, inverse system and meta-system is introduced. This idea is then illustrated for the identification of a symmetric signal found in a scanning radar system by way of a computer aided design containing a signal generator, a signal detector and an optimiser/controller.

Particular attention is paid to defining an association function from which the probabilities of success and error follow. The relevance of these techniques to symmetric patterns is also discussed.

3.1 A Cybernetic System

In order to assess the effectiveness of symmetric signal detection, a simulation was carried out by the author, which is summarised in Appendix 1. Before giving further details, it is useful to look at the general form of this simulation, as it may prove valuable to future research.

As can be seen from Figure 3.1, a cybernetic system appears to consist of three essential elements:

a) A System

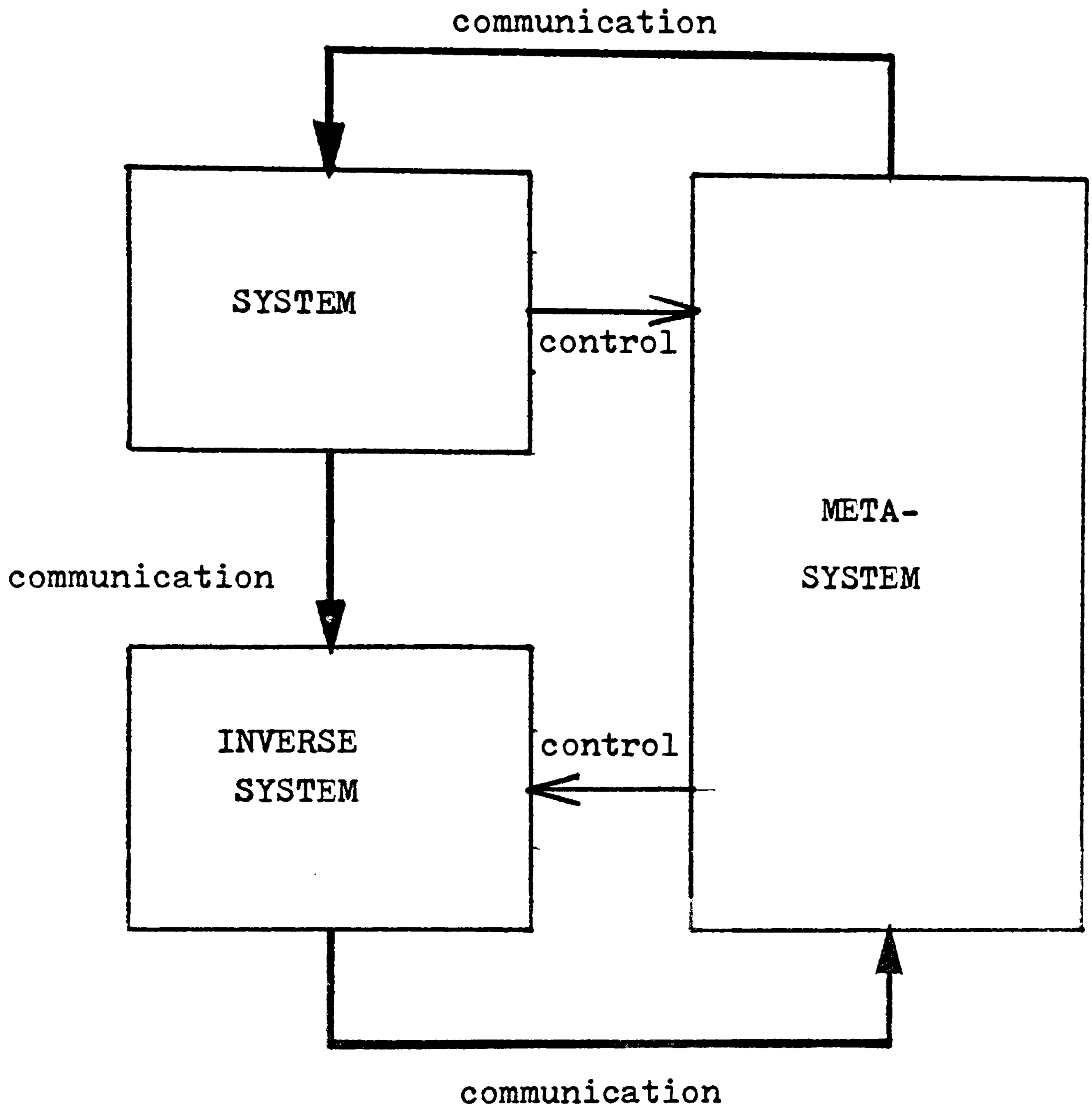
b) An Inverse System

c) A Meta System (which may have the elements a and b)

This form of recursive definition is common in mathematics/systems theory, but in order for the uninitiated to establish a contextual frame, the system may be associated with a pattern generator (eg the environment, a coder, a stimulus) the inverse system may be associated with a pattern recogniser (eg a computer, a decoder, an organism) and the meta-system with a comparator and optimiser (eg an operator, an observer, an experimenter).

The interconnexion lines being those of communication between the system and the inverse system and control lines between the meta-system and the system/inverse system.

Thus the meta-system may modify the behaviour of either the system or the inverse or both, thus directly affecting the communication link.



A CYBERNETIC SYSTEM FIGURE 3.1

However, any boundaries that may be identified in a real cybernetic system between system, inverse system and meta-system will normally be confused - no element being beyond modification. Nevertheless these mathematical/algorithmic simplifications do present a starting point and allowed the author to completely simulate a whole cybernetic system in an almost closed loop situation.

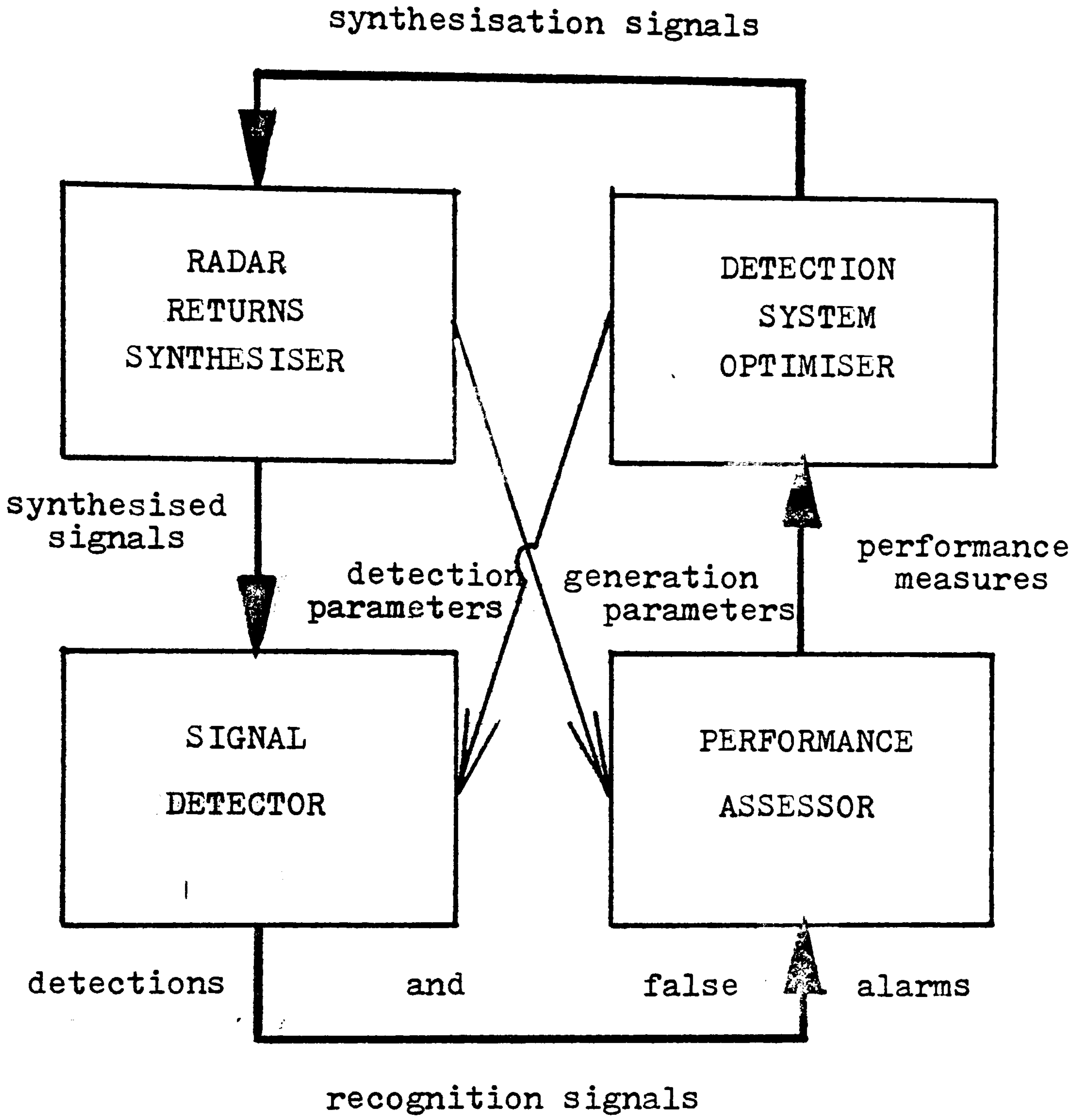
3.1.a The System

3.1.a.i A symmetric signal generator

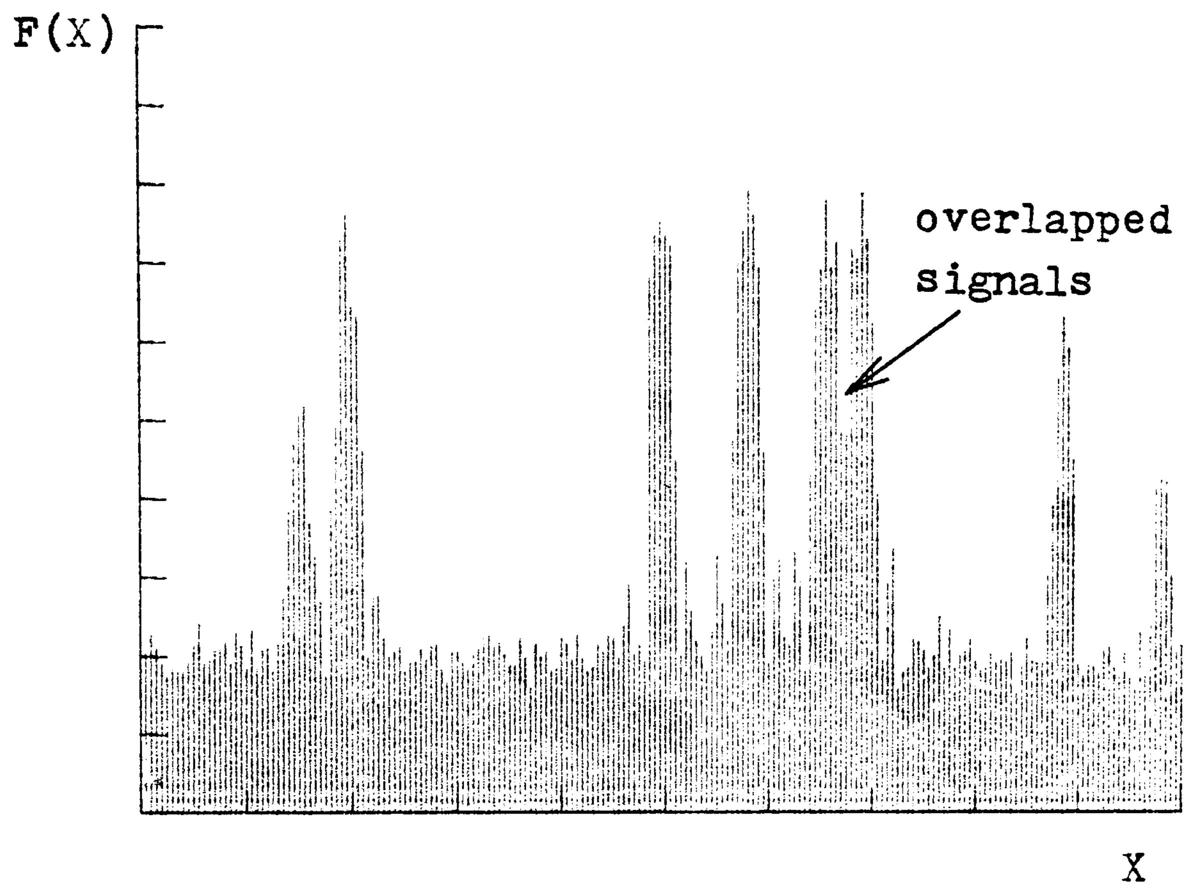
In order to generate locally symmetric signals randomly sampled, scaled and distributed in a background of white gaussian noise, the synthesiser shown in Figure 3.2 was used in a simulation by the author (see Appendix 2). This synthesiser is orientated around a random number generator which firstly generates truncated gaussian white noise by averaging a sequence of random numbers of rectangular distribution, then a randomly sampled $(|\sin x/x|)^n$ function was used as the symmetric signal since this is a good approximation to an antenna pattern; finally the noise and the randomly attenuated signals are randomly merged together and a decibel scale taken. Overlapping signals interfering with each other and causing asymmetries to result as are shown in Figure 3.3.

3.1.a.ii A symmetric pattern generator

Although a symmetric pattern generator has not been simulated, it is interesting to speculate how this might be achieved. Provided one is satisfied that a vector chain can be extracted from the input matrix, then the simulation could begin by



A SIMULATION OF A RADAR SYSTEM FIGURE 3.2



OVERLAPPING SYMMETRIC SIGNALS FIGURE 3.3

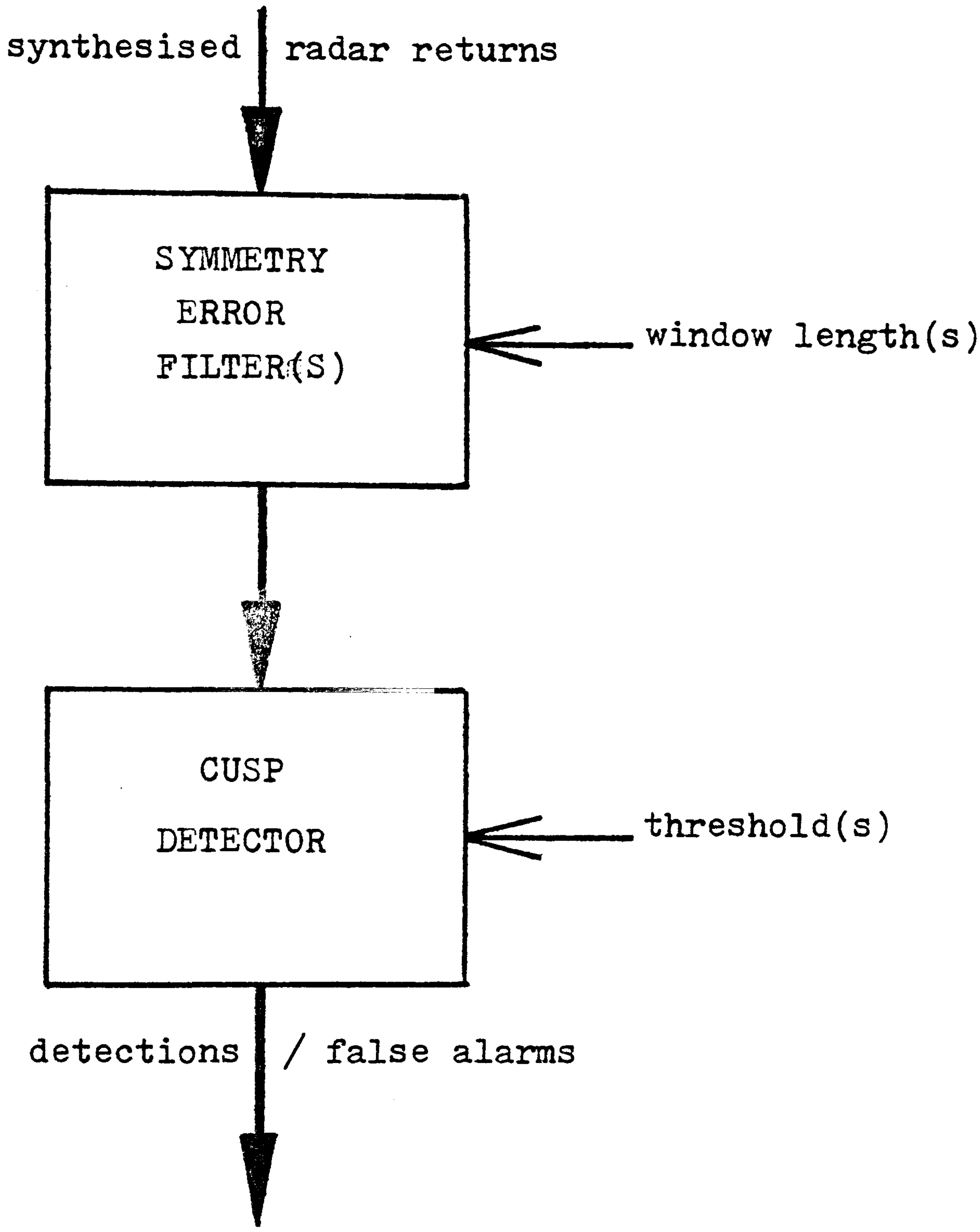
simulating a symmetric vector chain rather than the input matrix. This could be achieved by using rewrite rules similar to those described in the author's M.Phil. thesis. However, a computationally easier way is to store locally symmetric two-dimensional polar or cartesian equations and randomly sample these, ensuring closure of the chain by checking the vector sum is set equal to zero by adding the appropriate final vectors. In this way both globally and locally symmetric shapes might be generated. Checks to limit the length of the vector components could of course be included, the actual length of the chain being a bounded random number. Alternatively if the aim of the recognition system is to classify the set of alphanumeric characters, say, then these characters could be stored in vector form and random scaling, style variations in slope, proportion, blots, and granular noise could be included.

A serial-to-parallel conversion to return to the actual input matrix would of course be possible by carrying out the inverse operation to contour following and vector extraction if necessary.

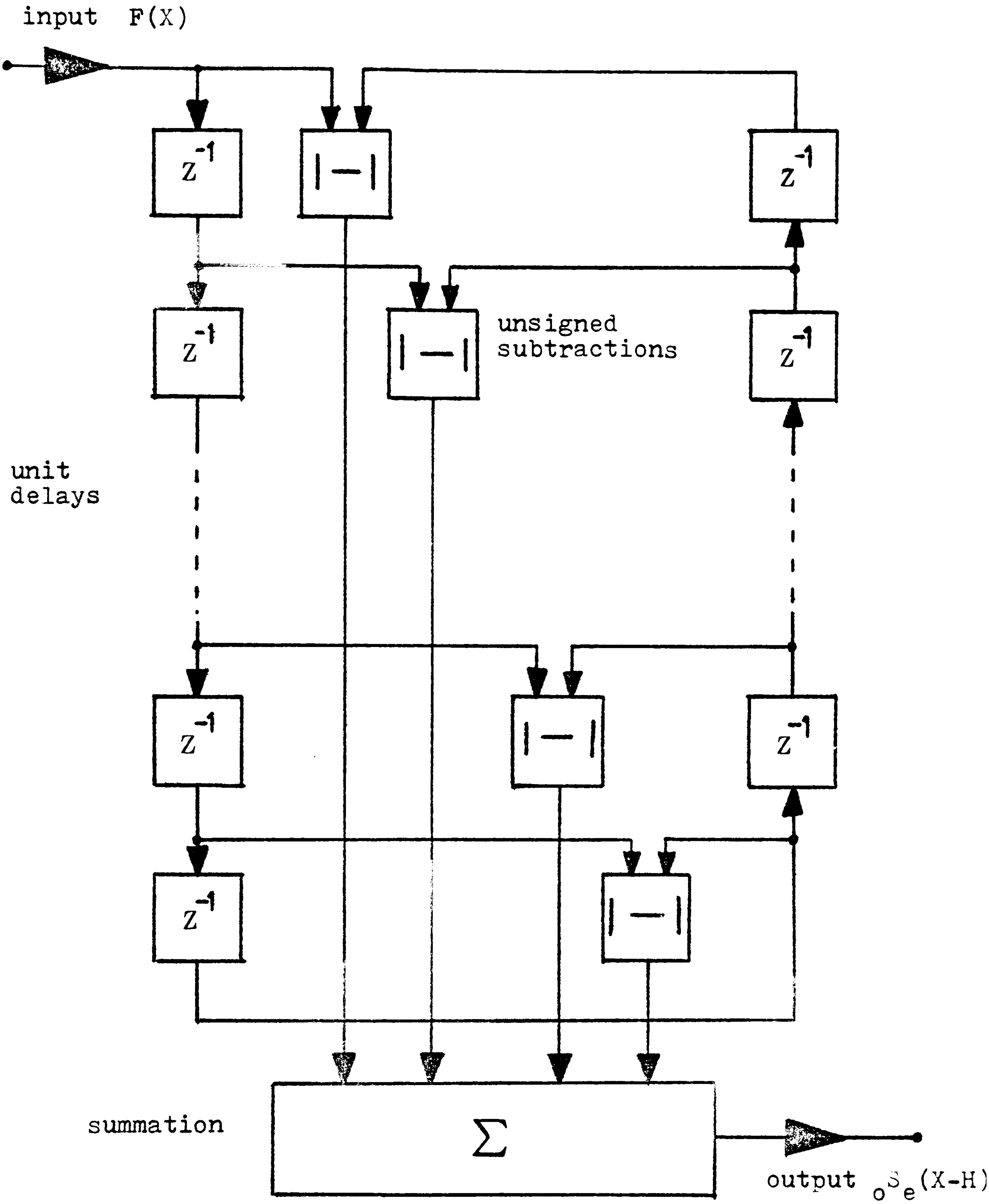
3.1.b The Inverse System

3.1.b.i A symmetric signal detector

Conceptually, the symmetric signal detector may be divided into two units: the symmetry error filter and the cusp detector as is shown in Figure 3.4. The mathematics has already been described in sections 2.2 and 2.4 and the actual algorithm is in Appendix Two. However, it is perhaps worthwhile to consider the symmetry error filter as a hardwire device as is shown in Figure 3.5.



SIGNAL DETECTOR FIGURE 3.4



A SYMMETRY ERROR FILTER FIGURE 3.5

Notice that the unit does not require any multiplication and has a minimum of storage.

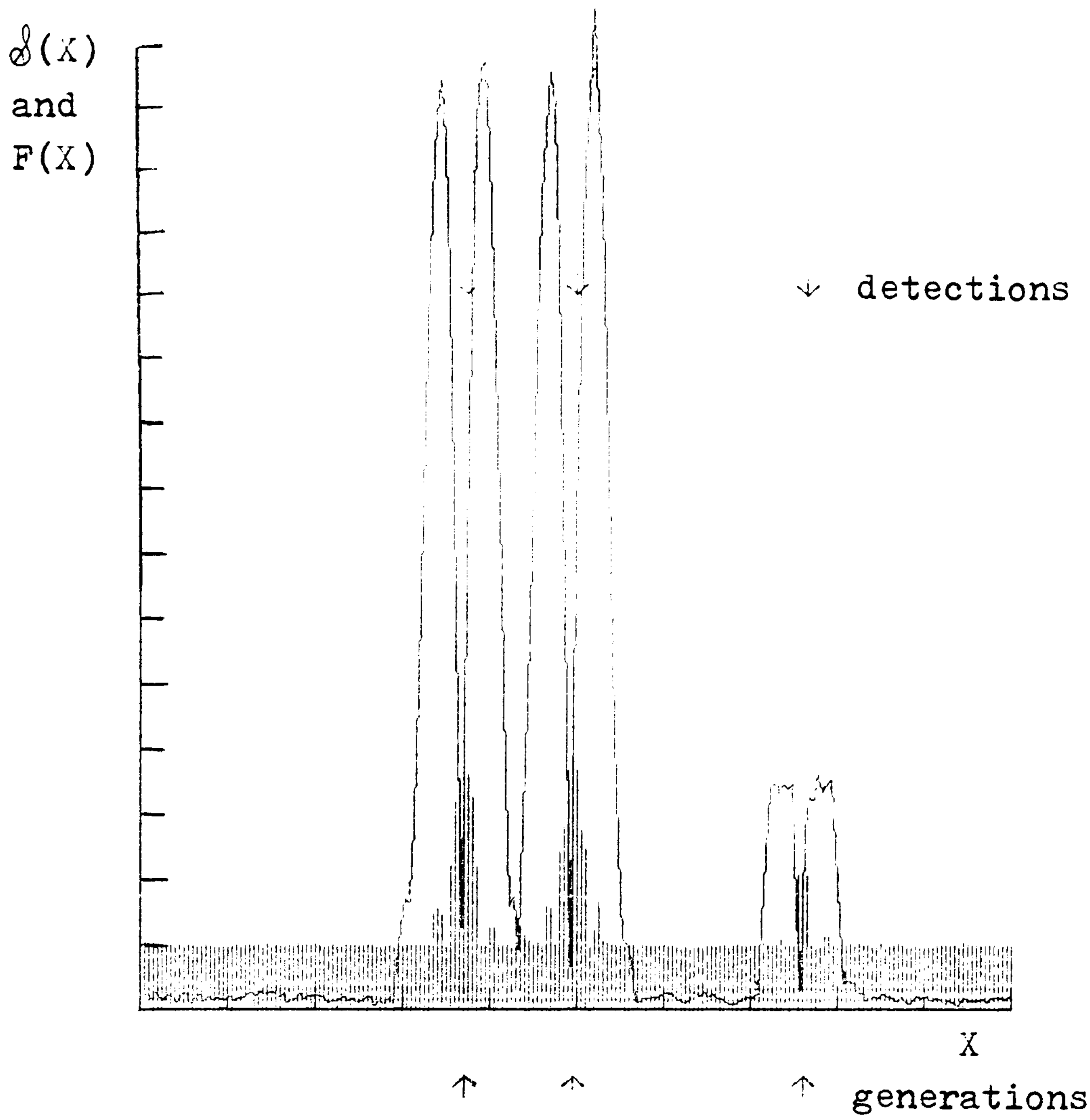
The results from this system when applied to antenna-like patterns described in section 3.1.a are shown in Figures 3.6 and 3.7, where it can be seen that the system does become confused when signals start to combine. However, if these signals are of similar extent, another unit can be included which separates them or gives an indication that there are more than two signals present (see section 1.1.e).

Plainly the choice of an optimum window length is not easy and in the last chapter ways of optimising this choice will be discussed.

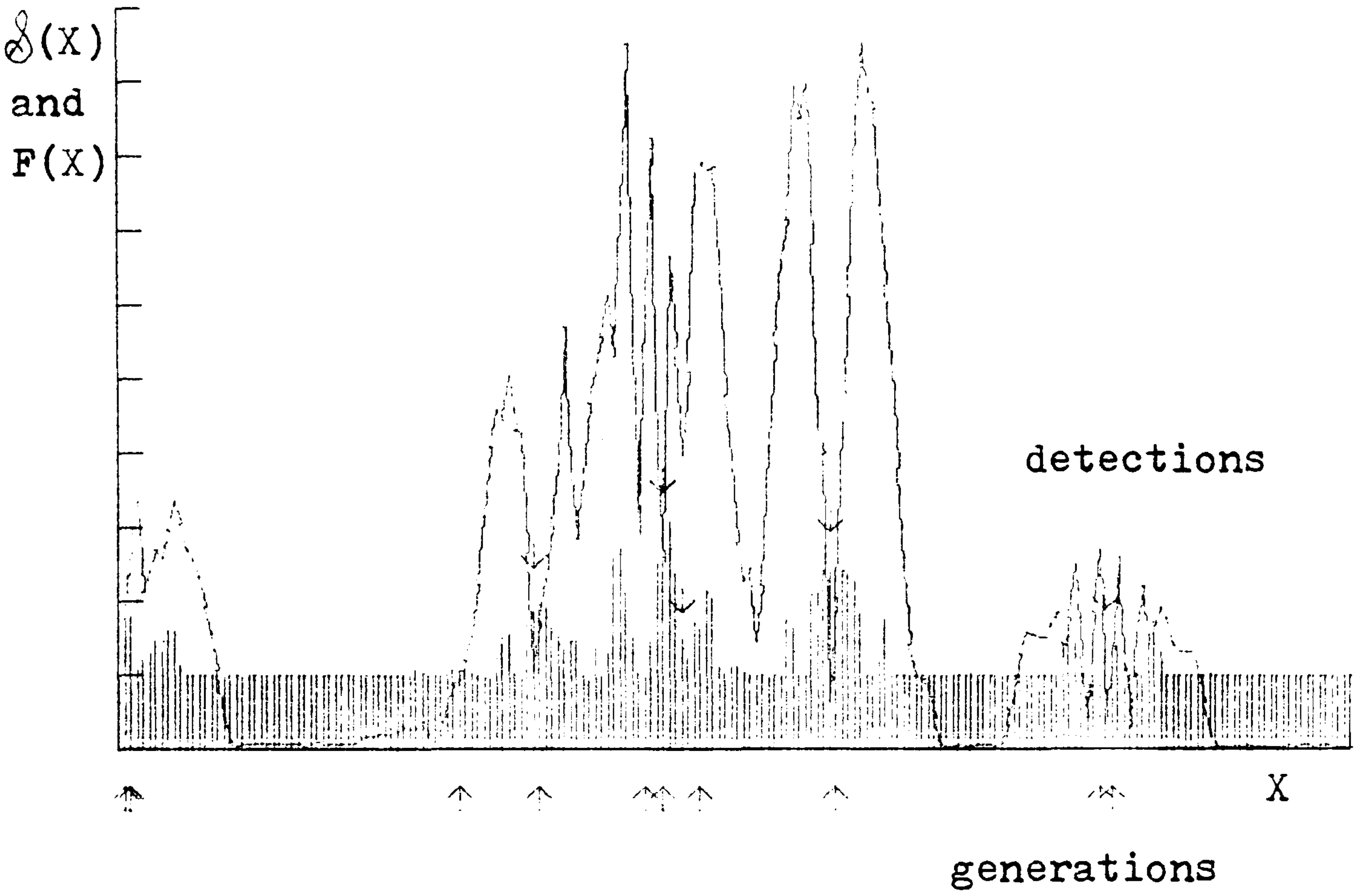
3.1.b.ii A symmetric pattern recogniser

As with the symmetric pattern generator, a symmetric pattern recogniser has not been simulated. However, it would not be unlike the symmetric signal detector in that it would first apply the error function described mathematically in section 2.2 and then search for nulls in this error function, there being no flat regions that are not points of symmetry.

This device would of course lend itself to much greater complexity since the occurrences of local symmetries would all contribute to the recognition of the overall shape.



SYMMETRY FUNCTION APPLIED TO MAINBEAMS FIGURE 3.6



SYMMETRY FUNCTION APPLIED TO SIDELOBES FIGURE 3.7

3.1.c The Meta-system

> "A meta-system is a system over and above the system itself. Its major characteristic is that it talks a meta language; this is a richer, better informed, way of talking than is available to the system lower down."
>
> Stafford Beer, 1975

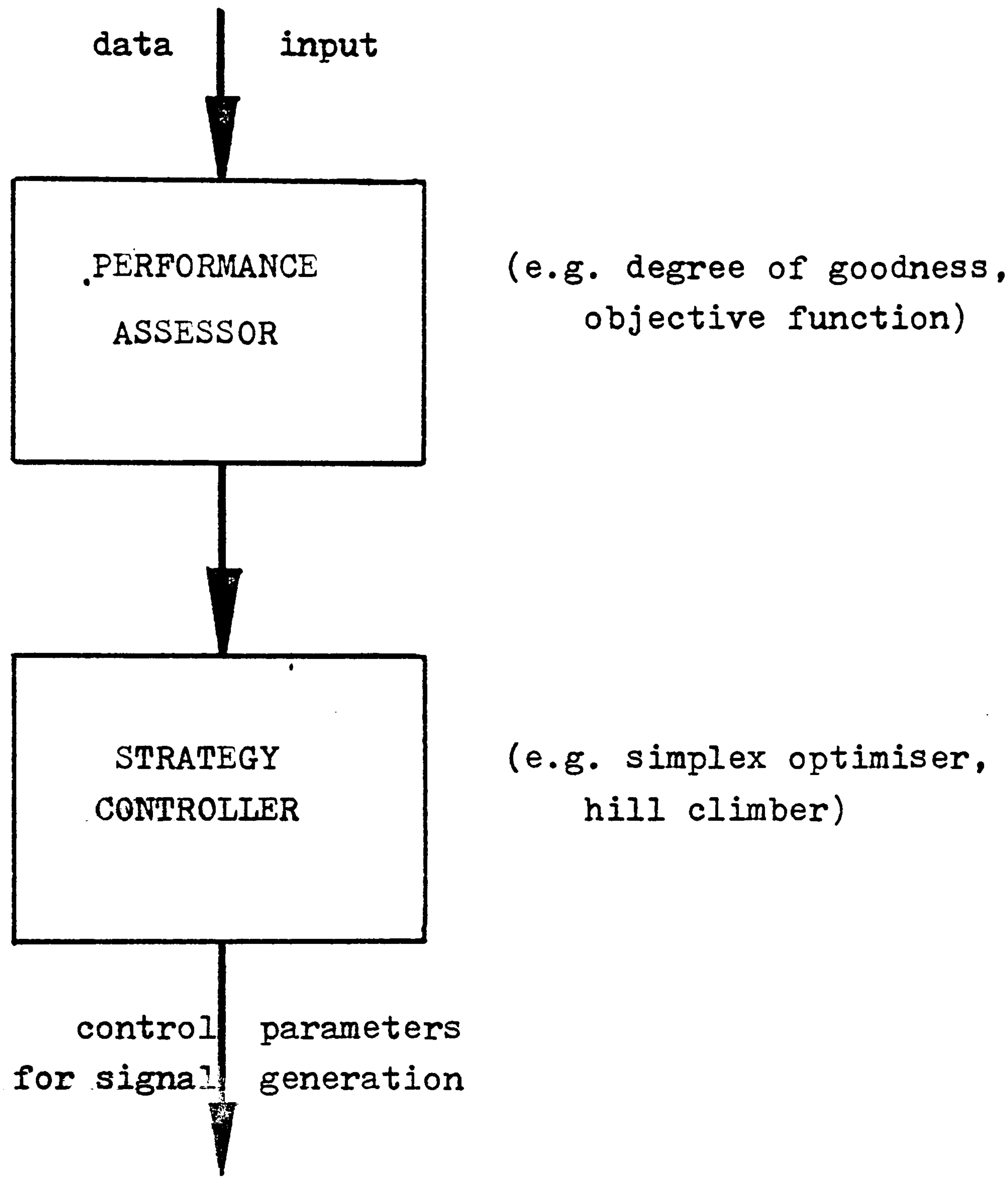
It can be seen from Figure 3.8 that the meta-system consists of two elements in the case of symmetric signal or pattern detection:

1 A Statistical Analyser (Performance Assessor)

2 A System Optimiser (Strategy Controller)

In cybernetic terms, if the meta-system is to effectively control it must be capable of directing as much variety as the system and inverse system can produce together, for ONLY VARIETY CAN ABSORB VARIETY - this is Ashby's Law of Requisite Variety (Ashby 1956; Beer 1975). This is not a difficult task for, although the variety of the system may be immense, the variety capable of being absorbed by the inverse system should be of a similar order and all the meta-system has to do is reduce the amount of uncontrolled variety in the system so that inverse system can be assessed and optimised. In Beer's managerial terms, the system may be the manufacturing process, the inverse system the market and the meta-system the management who hopefully can effect both system and inverse system by identifying the crucial parameters of control.

The mathematics of the statistical analyser will now be described, but the system optimiser will be left to Chapter 4, since this is worthy of the devotion of a complete chapter.



<u>A META SYSTEM</u> <u>FIGURE 3.8</u>

3.2 The Association Function and its Converse

Let the pattern generator produce an array I of dimension 1 to i of the co-ordinates of the axes of symmetry relative to some arbitrary but fixed element* in the 1-D function or 2-D vector chain. And, let the pattern recogniser produce an array J of dimension 1 to j of the co-ordinates of the identified axes of symmetry relative to the same arbitrary but fixed element in the 1-D function or 2-D vector chain. Thus there exist two arrays I and J, one describing the system and one the inverse system. In order to test the compatibilities of the systems, an association function "A" can be defined. This may be done by letting:

$$A(r)=I(r).\left\lceil\left|I(r)-J(t)\right| \leqslant \theta_A\right\rceil \quad \text{for all } 0<r\leqslant i \text{ and } 0<t\leqslant j \tag{3.1}$$

which will set all elements in I(r) that do not lie within a range $\pm\theta_A$ to zero in A(r). In a similar way, an inverse function A' may be generated by letting:

$$A'(t)=J(t).\left\lceil\left|I(r)-J(t)\right| \leqslant \theta_A\right\rceil \quad \text{for all } 0<t\leqslant j \text{ and } 0<r\leqslant i \tag{3.2}$$

which will set all elements in J(t) that do not lie within a range $\pm\theta_A$ to zero in A'(t). So that A(r) contains the co-ordinates of all symmetries that do associate in I(r) and A'(t) contains the co-ordinates of all symmetries that do associate in J(t).

* Chosen by the meta-system

Clearly the non-zero values (though not in order) in A and A' are identical. However, their converses A_c and A'_c are not identical, for:

$$A_c(r)=I(r).\left[|I(r)-J(t)|>\theta_A\right] \tag{3.3}$$

and

$$A'_c(r)=J(t).\left[|I(r)-J(t)|>\theta_A\right] \tag{3.4}$$

where

$$I(r)=A(r)+A_c(r) \tag{3.5}$$

and

$$J(t)=A'(t)+A'_c(t) \tag{3.6}$$

for all $0< r\leqslant i$ and $0<t\leqslant j$.

Thus the two most useful association functions are A(r) and $A'_c(t)$ for they perfectly describe the instantaneous performance of the system - A(r) relating to the number of symmetries recognised and $A'_c(t)$ describing the number of symmetries falsely generated.

3.3 The Probabilities of Success and Error

The association functions $A(r)$ and $A'_c(t)$ define the short term 'a posteriori' probabilities of success p_s and error p_e in a signal or pattern, that is:

$$p_s = 1/i \sum_{r=1}^{r=i} \left[A(r) \neq 0 \right] \tag{3.7}$$

and

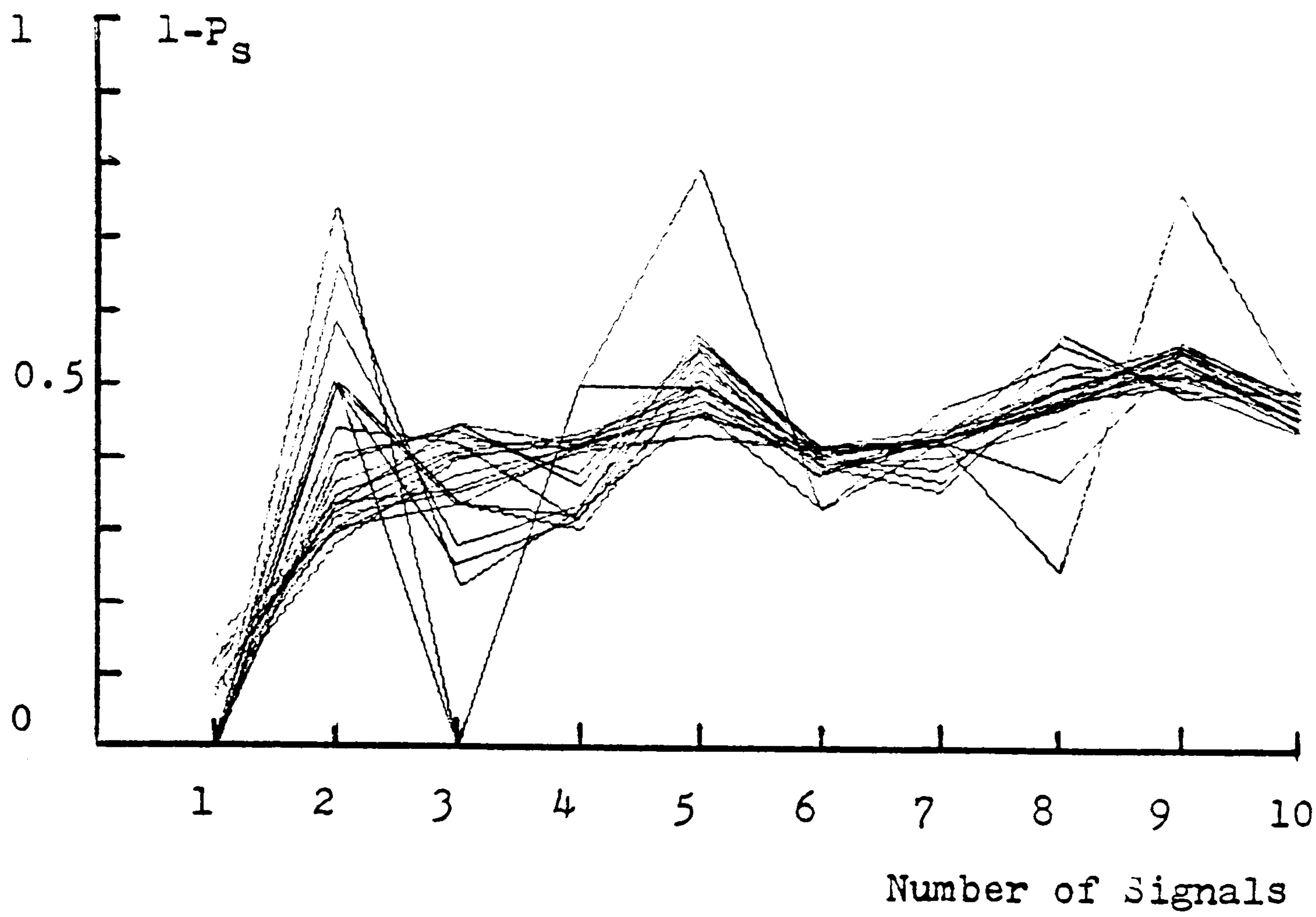
$$p_e = 1/j \sum_{t=1}^{t=j} \left[A'_c(t) \neq 0 \right] \tag{3.8}$$

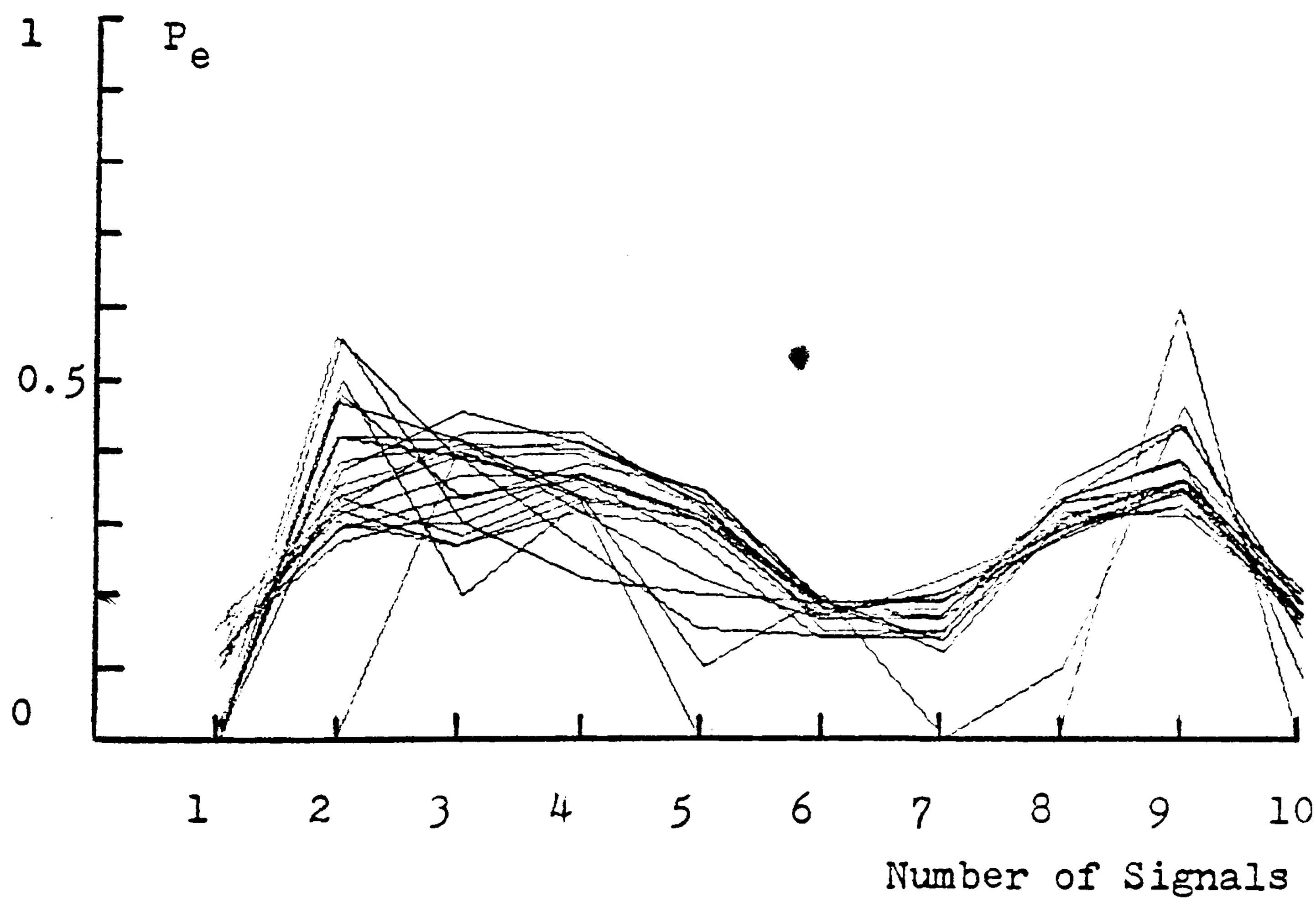
Note in particular that

$$0 \leqslant p_s + p_e \leqslant 2 \tag{3.9}$$

and that p_s is essentially statistically independent of p_e. Now these probabilities will of course change from signal to signal, pattern to pattern, and must therefore be averaged if accurate estimates are to be made of the overall system performance.

Some convergence curves are shown in Figure 3.9 that are taken from the simulation given in Appendix One. The long term 'a posteriori' probabilities of success P_s and error P_e being expressed as:





PROBABILITY CONVERGENCE GRAPHS FIGURE 3.9

$$P_s = 1/N \sum_{R=1}^{R=N} p_s(R) \tag{3.10}$$

and

$$P_e = 1/N \sum_{R=1}^{R=N} p_e(R) \tag{3.11}$$

CHAPTER FOUR

OPTIMISATIONS OF THE SYMMETRIC MESSAGE IDENTIFICATION PROCESSES

Foreword

The general problems of finding the optimal parameters for a cybernetic system are introduced and are associated with the special techniques of symmetric signal detection schemes established in the previous chapters.

Optimisation is defined for both globally and locally stable parameters and standard methods, such as the sequential simplex method, are linked with the computer aided design of Appendix 1.

The unusual tactics of recursively applying the symmetric error function to establish an objective function or alternatively using a stochastic objective function based on the probabilities of error and success are discussed.

4.1 Local and Global Optimisations

> "The statistical studies necessary to use a long past for a determination of the policy to be adopted in view of the short part are highly non-linear."
>
> Norbert Wiener, 1961

As has already been indicated, any optimisation process requires some criteria of goodness/badness by which to work and essentially consists of maximising/minimising an associated function (objective function) that embodies these qualities.

The identification of the parameters requiring optimisation in the cases of symmetric signal and pattern identification have already been made - they are the windows and thresholds associated with the extent of the symmetric error function and the thresholds of the detection processes respectively. Now this directly indicates that two forms of optimisation are considerable:

1 Local optimisation

and

2 Global optimisation

Local optimisation is applicable to the window length since this is unlikely to remain fixed for any length of time in a varied environment and should be matched according to the stimulus by a short loop path of low hysteresis based on the association function. However global optimisation of the threshold can be sought as a function of window length provided a statistical performance function such as the type described in Chapter Three is used and the pattern generator controlled to give only patterns of a certain extent. This long duration optimisation loop presents certain difficulties in practice with large classes of symmetric signals, since the convergence time may be very long.

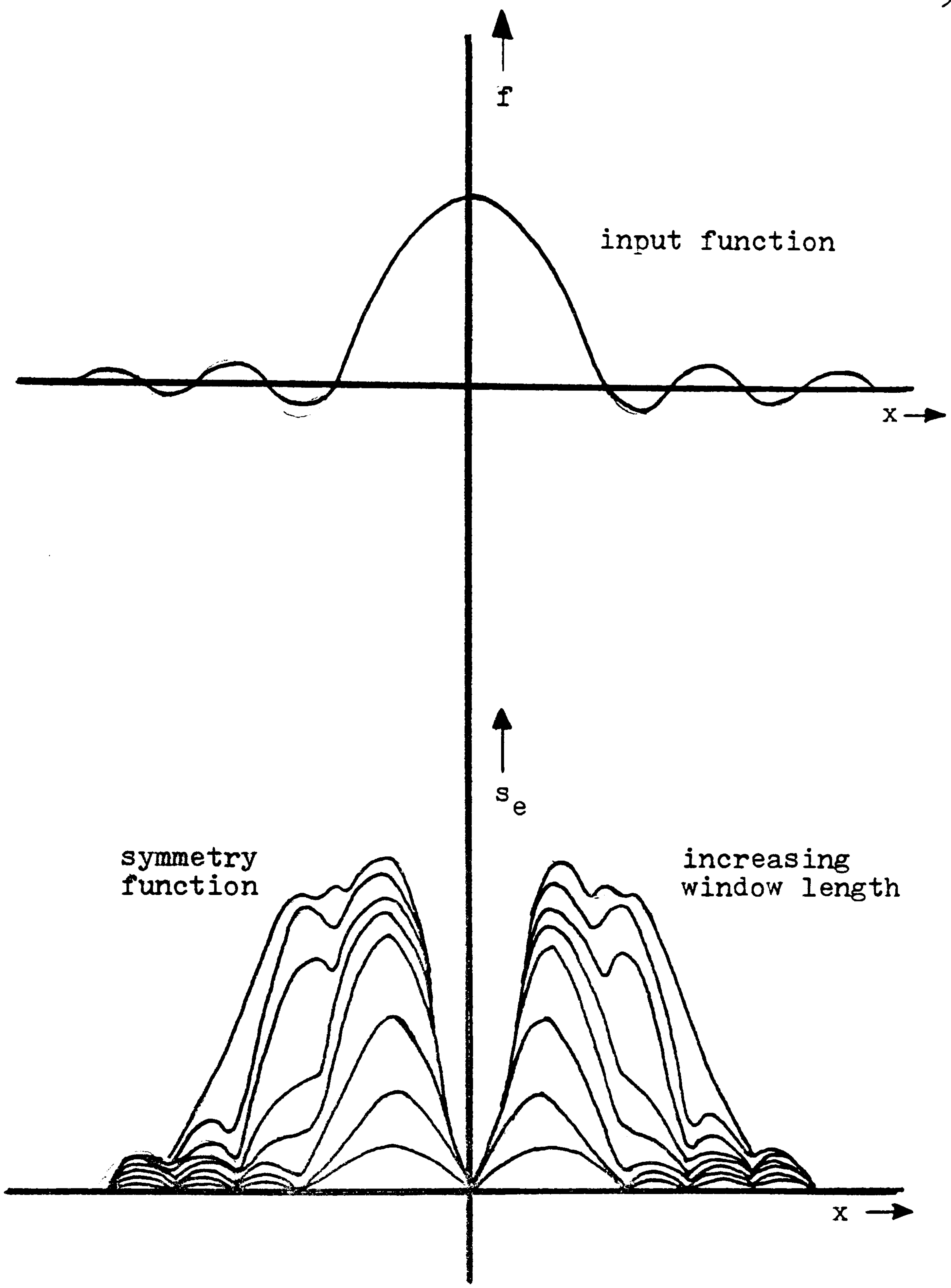
One must of course endeavour to keep a flexibility in the optimisation processes for although convergence of the statistical function may appear to have occurred, it may be of a non-stationary nature, especially in a non-homogenous environment.

4.2 Local, Uni-Modal Optimisation of the Window Length

If there is no way a meta-system can supply continuous true information between the pattern generator (the system) and the pattern recogniser (the inverse system) in the form of an error or control signal, then an open loop situation exists where the pattern recogniser must function as a closed-loop or autonomous system, only receiving unverified input from the pattern generator.

In order to combat this situation, the objective function for the pattern recogniser must be totally self-sufficient in order to carry out local optimisation or be able to go into a training mode whereby a temporary inclusion to the loop of the meta-system be agreed. For example, in considering the optimisation of the window length of the symmetry error function in the example of Appendix 1, a possible objective function could be based on the number of symmetric detections as a function of window length (see Figure 4.1). Plainly this objective function requires minimisation, since symmetric detections are most likely to occur when the symmetrically placed side-lobes are detected and clearly this is not desirable.

Other a priori information is available, such as the fact that first sidelobes are greater than ydB down on the main beam and so on, that can all be included to minimise the chance of the detection system being in error or failing to detect a signal. A judicious increasing of the window length as the possible aircraft approaches and more of the signal rises above noise should tend to minimise the just-described objective function.



EFFECTS OF WINDOW LENGTH FIGURE 4.1

Another checking procedure might be to try a predict where the next detection should take place and then if actual detections are consistent with sidelobes, suppress the outer detections.

However, the intrinsic simplicity of the symmetric error system is now being lost and the designer must question his own skill as a super-meta system in installing this form of local optimisation.

Notice that in this situation the optimisation is resting heavily on the 'a priori' information which must have somehow been gathered by the designer.

4.2.a A Recursively Applied Symmetric Objective Function

In optimising a single parameter whose objective function is not analytic, a number of tests or experiments must be performed to determine when the function is either a maximum or minimum for a set of given values of this parameter.

For example, in the case of the detection system for the axis of symmetry of an antenna pattern with multiple sidelobes, the objective function should be evaluated for various window lengths.

An elegant way of defining the objective function O(H,N) for this rather special example is to apply the symmetry function and detection function to the detection function D(H,N) over all 'N' that is

$$O(H,N) = \sum_{R=0}^{R=N} D(S(D(H,R))) \tag{4.1}$$

the extent of S being maximal and D maximally sensitive.

Clearly O (HN) should be so adjusted to maximise the number of asymmetric detections, that is, minimise the number of symmetric detections, since all symmetric detections must be suspect. This form of recursive testing does of course have its problems in that - how are the reapplied functions (that is the objective functions) optimised? - but it seems that their optimisation is not as critical since there is a movement towards a more idealised situation (ie situation of less noise).

4.2.b Different Single Dimensional Search Techniques

Methods of choosing 'H' to quickly arrive at an optimum value are numerous; for example, the Fibonacci search which uses only one experiment for each cycle (except the first, when two are used) makes use of the placement of the experiment in previous cycle and can be seen to be optimal in its number of tests.* If there are only a few values of H to pass through, a straightforward linear search may prove desirable, due to its simplicity, especially if only small adjustment in H is required.

* Fibonacci numbers have the property $F_N = F_{N-1}+F_{N-2}$ $(F_0=F_1=1)$
To find an optimum for a unimodal function within the interval of search, L_k, L_k may be selectively reduced (initially on both sides k=1) by

$$l_k = \frac{F_{N-(k+1)}}{F_{N-(k-1)}} L_k$$

on each successive iteration of k, since the remaining point will be correctly placed for the next test and so on until N=k (see Beveridge and Schechter 1970).

Although Fibonacci numbers date to Leonardo de Pisa (13th century) (and before in the form of 'aurea sectio') this technique was only discovered in 1953 by Kiefer.

4.3 Global, Uni-modal Optimisation of the Thresholds

Suppose there exists, or can be made to exist, a continuous supply of true information to the pattern recogniser, then a closed system exists where the performance of the pattern recogniser may be made depend on the meta-system.

4.3.a A Stochastic Objective Function

In practice one way of achieving this absolute knowledge is for the meta-system to take control of the pattern generator so that only signals of a certain type may be produced. By taking this action, it is possible to establish statistics as to the performance of the pattern recogniser. This serves two useful functions: the system need not be optimised on real data and the amount of prototype design carried out by the designer of the super meta-system is limited since all the work can be performed through computer simulation. This type of procedure was tried for the thresholds in the symmetric signal detection process (see Appendix 1) when the objective function was simply defined as the sum of the average probabilities of success and of no error over a stepped number of signals, that is:

$$O_s(n)=1/2n \sum_{i=1}^{i=n} (P_s(i)+(1-P_e(i)) \tag{4.2}$$

the objective function 'O_s' is thus of a stochastic nature and not directly specifiable by an analytic curve but derivable by the control of the meta-system.

(Continued page 107)

4.3.b Different Multi-dimensional Search Techniques

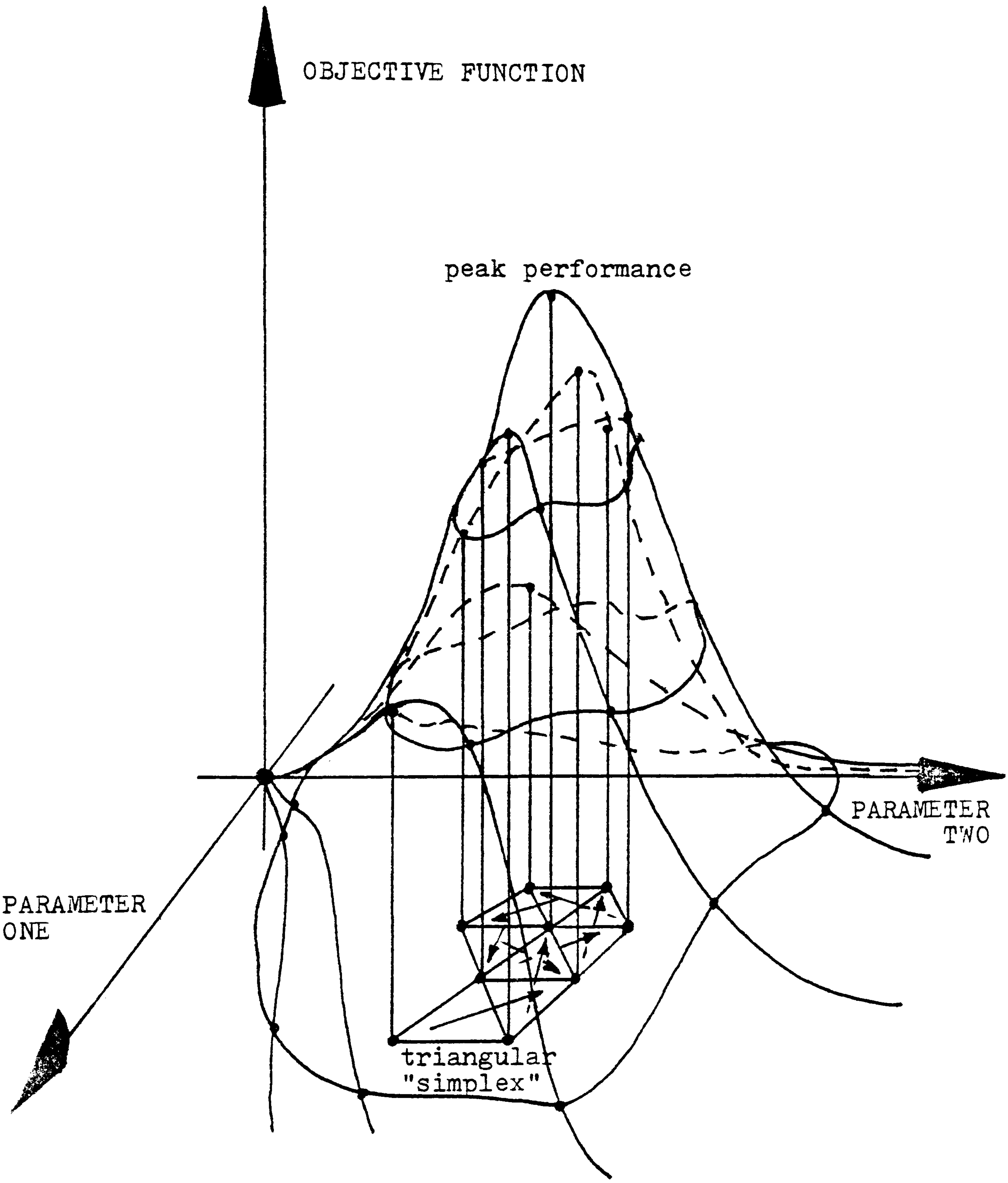
"Simplex munditiis."

Horace

In optimising a set of parameters that are not analytically related to the chosen objective function, a number of tests must be performed to determine when the function is either a maximum or minimum for given values of these parameters. This is many times harder than the one-dimensional case, since the number of degrees of freedom is greater and optimisation of one parameter for the others held fixed does not normally lead directly to an optimum.

There are, in fact, many ways to optimise a set of parameters when the objective function is essentially uni-modal in the region of interest, as is the case with expression 4.2 of the last section, most of them being of the hill climbing/descending strategy.

As the thresholds implicit in expression 4.2 are continuously varying unconstrained parameters, the simplest method appeared to be the sequential simplex technique, a procedure for which - written by the author - is detailed in Appendix 1. This technique is well known and therefore only warrants a brief description here. Essentially the algorithm is to generate a set of n+1 equidistant points in the n dimensional parameter space that spans all the dimensions (eg in 2D space - the vertices of an equilateral triangle, in 3D space - the vertices of a tetrahedron), known as "a simplex". The values of objective function



A 2D SIMPLEX OPTIMISATION

FIGURE 4.2

at these n+1 points are then evaluated and the minimum valued vertex (for maximisation) changed in favour of its reflexion defined by its twin hyper-sphere which passes through the remaining n points and is of the same radius. The new point is then used to evaluate the objective function and the new minimum vertex reflected in the way that has just been described.

A mental picture can be built up by considering the two dimensional case which is illustrated in Figure 4.2. This process continues until a point is cycled round when the simplex's circumscribing radius is either reduced or the optimum is defined for this point. Notice that, as with the Fibonacci search, only one point is added per decision cycle. In order to avoid being caught on a ridge, a test must be included to ensure an already visited point is not returned to and the second smallest value chosen instead. Also if the objective function is stochastic, then the procedure should include a more liberal test for a maximum than a single cycle round, the objective function being re-evaluated at these points until the experimenter is satisfied that a maximum due to local perturbations has been ruled out and a true optimum is arrived at.

This technique was in fact applied to the thresholds and a table of optimisation is shown in Figure 4.3. Notice that the optimisation did not terminate at an optimum - in fact the programme had to be curtailed since the maximum allowable computer time (2 days) had been exceeded. This is plainly a disadvantage with this sort of approach especially when the objective function is stochastic and

Objective Function	Threshold 1	Threshold 2	Threshold 3
.57	0	1	0
.55	.94	1.24	.24
.56	.24	1.94	.24
.59	.24	1.24	.94
.70	-.62	1.54	.54
.59	-.50	.58	.75
.57	-.99	.85	-.07
.52	-.57	1.69	-.44
.57	.18	1.97	.14
.62	.28	1.33	.90
.61	-.41	.61	.82

program curtailed

System parameters: f=20;m=8;MN=10;k=0;g=8;e2=.1;c=.1;
d=1;M=2;u=0;v=1;w=0;a=1;b=10;s=1,
(see Appendix Two for definitions of parameters)

A TABLE FOR SIMPLEX THRESHOLD OPTIMISATION FIGURE 4.3

requires considerable time to be evaluated (10 minutes per point on a slow machine of cycle time 3μS). As the value of thresholds were not required for any piece of apparatus at this particular point in time, this type of optimisation was not continued, but judged somewhat impractical with the present objective function and type of computer. However, the principle appears to be sound.

Other more complicated optimisation techniques that work within constraints such as the complex method of Box were considered, but these appeared to offer no obvious advantages except the region of search is constrained.

CONCLUSIONS - SOME GENERALITIES AND SOME PARTICULARS

> "Symmetry, as narrow or as wide as you may define its meaning, is one idea by which man through the ages tried to comprehend and create order, beauty and perfection."
>
> Hermann Weyl

The thesis can now be assessed as a cybernetic philosophy and thus the previous four chapters drawn together in a coherent way, in so doing highlighting what has been answered and what remains to be questioned.

The main theme of the text has been that of symmetry, which has been shown to be of essential importance to cybernetics in that it represents a highly ordered situation found naturally throughout our chaotic universe. Although the information content of a symmetric message is lower than an asymmetric one, since it repeats itself and thus is redundant, there is something aesthetically beautiful and intriguing about its generation and recognition that had led man to build his knowledge around the concept of symmetry.

The mystery of symmetry has not really been unveiled here, only definitions for its generation and recognition as mathematically formulae have been given. The significance of these formulae can only be judged in the context of their application to reality. Several examples of how these definitions apply to one and two dimensional symmetric situations have been given. The crucial point in these examples is that an imperfect world has had to be interfaced with the perfection of mathematics and some ad hoc decisions made as to the presence of symmetry.

The decision process is one of dividing noise from signal and can only be achieved by comparison with known properties of the signal and the noise. Symmetry is a very extraordinary event in Gaussian noise and as such has been shown to be easily separable. But at what point symmetry becomes asymmetry is a question that cannot be answered by theory - an arbitrary decision has to be made by the recogniser as to the fractional distortion in perfection that can be tolerated. This may be a fixed level (ie a threshold), but may also be a function of the noise, since what is classified especially in the 2D pattern recognition case is normally in some sense a nearest match to what is or has been available.

Thus a major part of this text has concerned itself with the determination of threshold levels that may be derived from statistical considerations of the association function and further kn own properties of the symmetric signals and patterns. The optimisation processes discussed have been indicated to be only as good as their objective functions and the automatic choice (ie the rules of choice) of these functions is by no means solved. However, one aspect that has emerged that is worth further study is the optimisation of the objective function by a recursive optimisation algorithm.

There remain two other properties of symmetry to be emphasised. The first is that symmetry is conserved only under convolution with other symmetric signals - under the more straightforward addition of two symmetric signals asymmetry can be introduced and inverse operation of the extraction of the two composite symmetric signals is only achievable if they are of the same extent. Secondly, the auto-correlation of any asymmetric signal produces an even symmetric function. That is to say, symmetry tends to be produced when a waveform "interferes" with itself, this is manifest in interference/diffraction patterns found in optics and crystallography.

Certain offshoots of this study of symmetry as a cybernetic concept have also emerged. For example, the idea that a cybernetic system is composed of a pattern generator (ie a system) a pattern recogniser (ie an inverse system) and an optimiser (ie a meta-system) is a concept that is implicit in most cybernetic works, but the author believes that here has it been made explicit through the concept of symmetry and its recognition. The application of this concept in a closed simulation of a radar system has confirmed the usefulness of this approach.

A technique that has stemmed from the author's earlier work is that of applying contour following by logical differencing nets and connectivity transition grammars to 2D symmetric situations. This has allowed for the recognition of locally symmetric patterns surrounded by global asymmetry and a topological connectivity invariance has been described.

But what of future research into symmetry as a cybernetic concept? In this thesis the primary concern has been with what might best be described as spatial symmetries. There does, however, exist a slightly different universe where symmetry begins to lose its spatial connotations. For example in a conversation the concept of reciprocity (a type of symmetry) plays a crucial role in synchronisation. As Colin Cherry remarks:

> "A conversation forms a two way communication link; there is a measure of symmetry between the parties, and messages pass to and fro."

That is to say A's utterances reflect B's utterances and vice versa. Here a grammatical system must be considered that embodies game theory and decision making, the predication and syntactic processes obeying certain rules of symmetry (eg reflexion, rotation, etc).

Actually it is now possible to deduce a general class to which only symmetric messages belong, that is irrespective of spatial and temporal considerations and depends only on numerical order.

Consider a message 'M' represented as a series of symbols:

$$M = \{M_1 M_2 M_3 \ldots\ldots\ldots M_n\} \qquad \text{C1}$$

then a symmetric equivalence $\underset{s}{\sim}$ between two symbols may be represented thus:

$$M_p \underset{s}{\simeq} M_q \quad / \quad p,q \in \{1,2 \ldots n\} \qquad \text{C2}$$

and implies:

$$M_q \underset{s}{\simeq} M_p \quad / \quad p,q \in \{1,2 \ldots n\} \qquad \text{C3}$$

Such a symmetric equivalence may be established by testing the symbols for numerical equivalence of properties, either by:

$$P(M_p) = P(M_q) \tag{C4}$$

where P is an <u>absolute</u> property operator or by:

$$M_p = \varrho(M_q) \tag{C5}$$

where ϱ is a <u>relative</u> property operator with a simple inverse ϱ^{-1}, such that:

$$M_q = \varrho^{-1}(M_p) \tag{C6}$$

<u>Reflexional Symmetry</u>

Now let p_r^+ be a forward counting sequence:

$$p_r^+ = p, p+1, p+2 \ .... \ p+r \tag{C7}$$

and q_r^- be a backward counting sequence:

$$q_r^- = q, q-1, q-2 \ .... \ q-r \tag{C8}$$

then a symmetric equivalence $\underset{s}{\sim}$ between two sub-messages may be written thus:

$$M_{p_r^+} \underset{s}{\sim} M_{q_r^-} \quad / \quad r \in \{1, 2, \ .... \ n\} \tag{C9}$$

and means that:

$$M_{p+r} \underset{s}{\sim} M_{q-r} \qquad \text{for all } r \tag{C10}$$

and further that:

$$P(M_{q-r}) = P(M_{q-r+1}) = \ \dots \ = P(M_{p+r-1}) = P(M_{p+r}) \tag{C11}$$

that is, each symmetric symbol equivalence should share some common property. If p=q then a contiguous sub-symmetry exists and if M is closed, that is:

$$\left\{M_1 M_2 \ldots M_n\right\} = \left\{M_{1+r}, M_{2+r}, \ldots M_{n+r}\right\} \qquad \text{C12}$$

for all integer 'r'

the symmetry may become total by letting 'r' completely span 'n'.

Rotational Symmetry

In a similar way, rotational symmetry may be defined thus:

$$M_{p_r^+} \underset{s}{\sim} M_{q_r^+} \; / \; r \in \left\{1, 2 \ldots n\right\} \qquad \text{C13}$$

and means that

$$M_{p+r} \underset{s}{\sim} M_{q+r} \quad \text{for all } r \qquad \text{C14}$$

and further that:

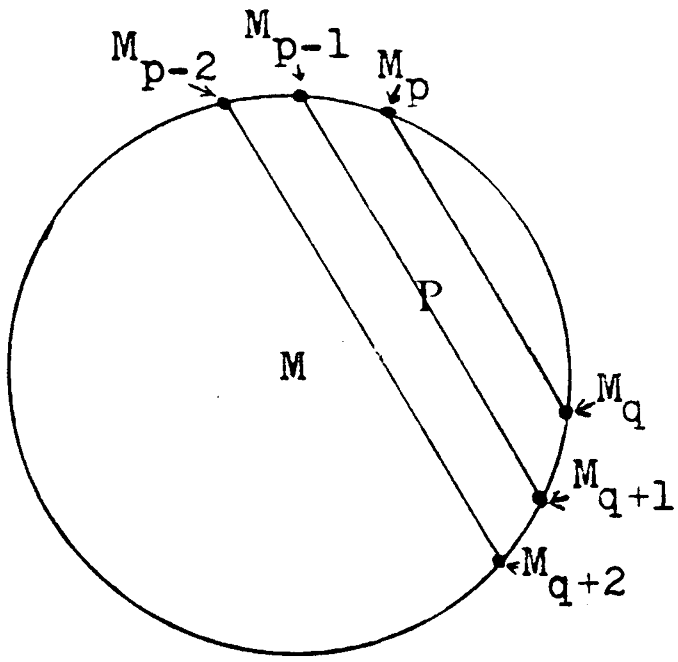
$$P(M_p) = P(M_{p+1}) = \ldots = P(M_{q+r}) \qquad \text{C15}$$

Translation Symmetry

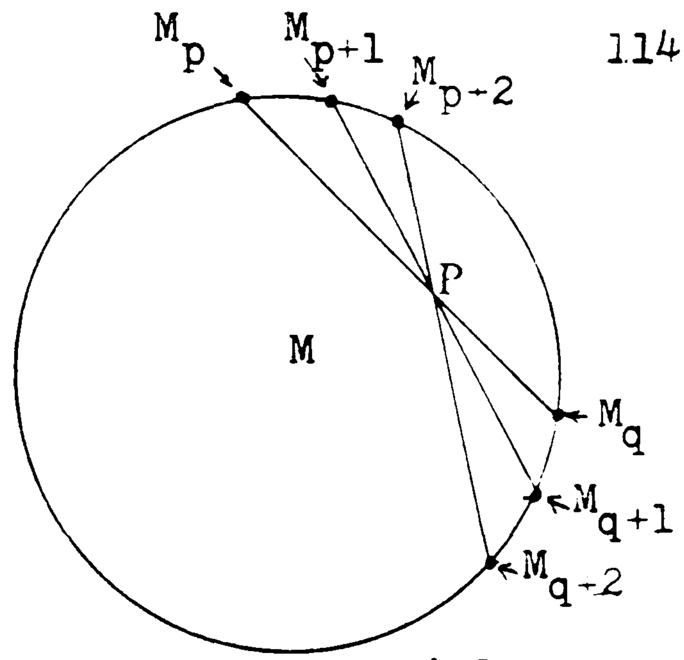
If two messages M' and M'' are defined to exist, which is possibly an unnecessary distinction, then a translation symmetry is simply:

$$M'_{p_r^+} \underset{s}{\sim} M''_{p_r^+} \qquad \text{C16}$$

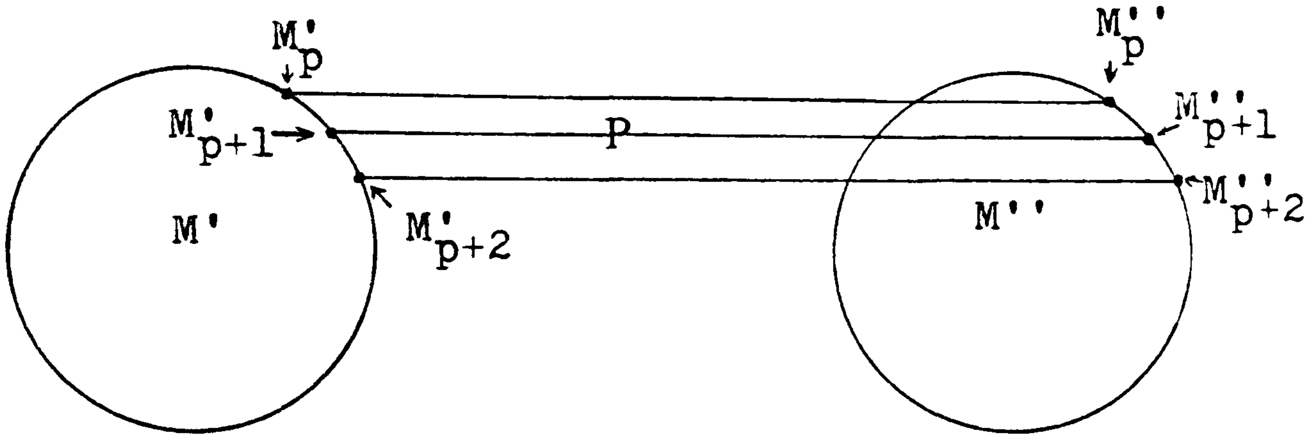
However, the indexing in M' and M'' may be different and the concepts of rotation and reflexion between messages would have to be introduced. The previous definitions are summarised by Figure C1.



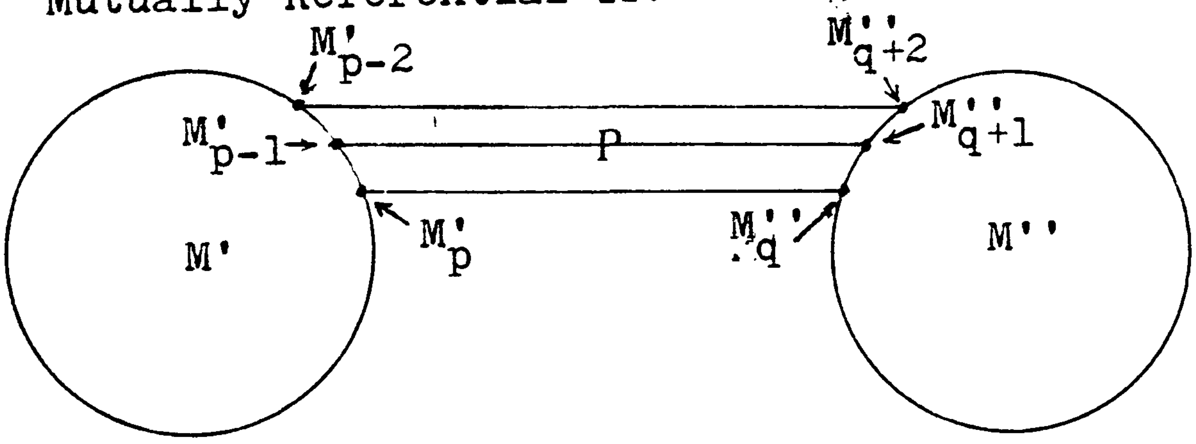
Self Referential Reflexional Symmetry



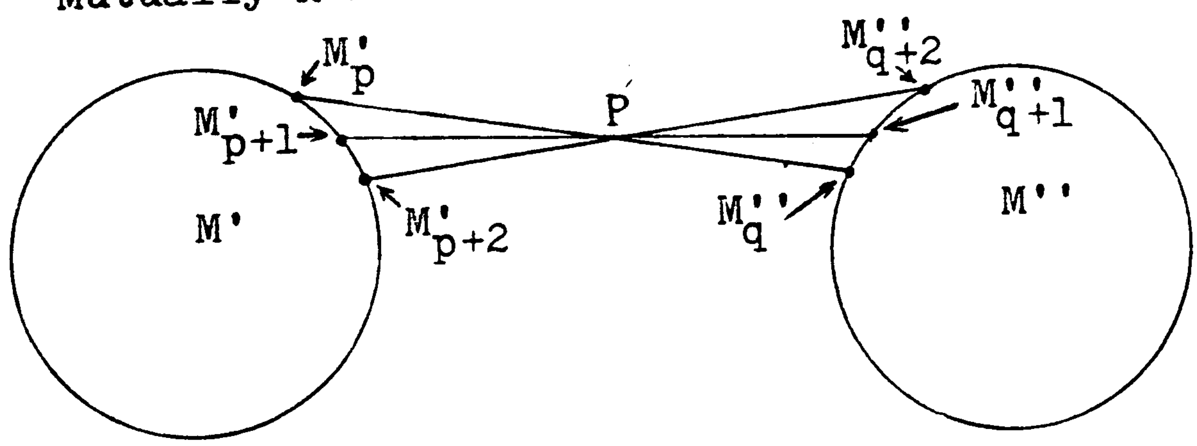
Self Referential Rotational Symmetry



Mutually Referential Translational Symmetry



Mutually Referential Reflectional Symmetry



Mutually Referential Rotational Symmetry

ABSTRACT SYMMETRIES FIGURE C1

These general definitions appear to agree with those of the atomic physicist where new types of symmetry are being discovered. In Abdul Salam's paper entitled 'Symmetry concepts and fundamental theory of matter' he asserts:

> "I felt that if nature must sacrifice the space-reflection symmetry principle it would do so only if this principle conflicted with some symmetry principle aesthetically even more appealing".

In fact in 1957 it was confirmed by Wu and Lederman that the neutrino's rest mass was zero and that no left-spinning neutrinos exist. Fortunately a more powerful symmetry concept has emerged, known as γ_5 symmetry, which can only be understood mathematically according to Salam (unfortunately he declines to explain the principle). However, he does go on to argue that the symmetry principle has led from protons to anti-protons (Dirac 1928) all the way through to the as yet unconfirmed 'quarks'. He remarks -

> "No mean triumph for the symmetry principle".

Also in the theory of computation and meta-mathematics, the symmetry principle is manifest. In 1943 Emil Post wrote a paper arguing that 'expressions' of a logical system or language, whatever else they may seem to be, are in the last analysis nothing but strings of symbols in some finite alphabet and thus even the most powerful mathematics or logical system is ultimately nothing but a set of rules that tell how some string of symbols may be transformed into other strings of symbols (Minsky 1972). He goes on to discuss rules he terms 'productions' which specify how to generate new strings of symbols from old ones. Although productions do not require the concepts of machines, they can be related to effective procedures and thus associated with the general theory of computation.

A pertinent system employing productions is given by Minsky:

Alphabet a, b, c

Axioms a, b, c, aa, bb, cc

Productions $\emptyset \longrightarrow a \emptyset a$

$\emptyset \longrightarrow b \emptyset b$

$\emptyset \longrightarrow c \emptyset c$

where the above can generate all possible palindromes of alphabet a, b, c and by employing the inverse principle recognise all possible palindromes (this is not true for finite state machines - simply because they are restricted to a finite count).

Thus the definitions of symmetry given earlier may be changed into productions. However, as a practical technique for recognising symmetry, one of elimination, is not the best - normally short cuts have to be taken, especially in finding the common property.

At a different level the construction of systems that reproduce themselves symmetrically may be worth investigation. For instance, when a living cell divides, it does so symmetrically, that is, in brief, the DNA helices split down the centre and take on their appropriate partner bases until complete duplication of the hereditary information is achieved when the outer membrane can contract around the separating nucleii to form two distinct cells. The question should therefore be asked that, given the self-reproducing models

$C_{1,n}$	$C_{2,n}$	$C_{3,n}$
$C_{8,n}$	$C_{9,n}$	$C_{4,n}$
$C_{7,n}$	$C_{6,n}$	$C_{5,n}$

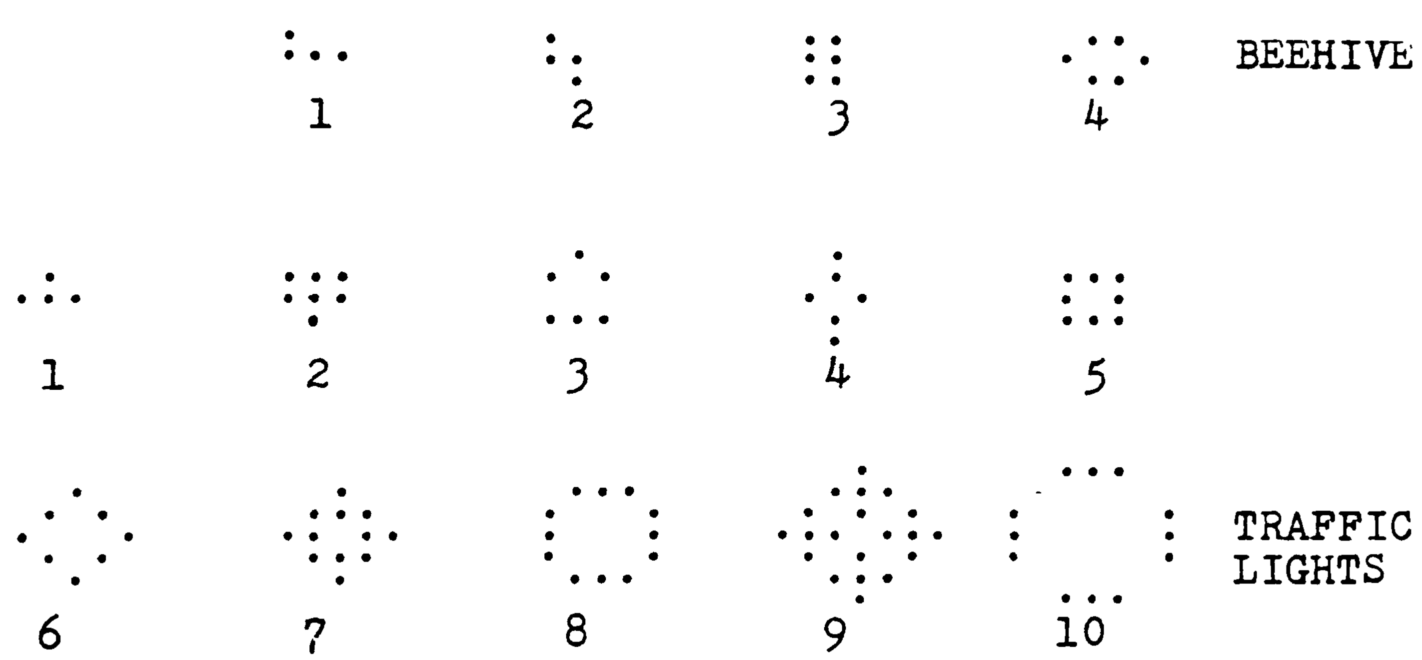
if $C_{9,n}=1$ and $2 \leqslant \sum_{r=1}^{8} C_{r,n} \leqslant 3$ then $C_{9,n+1}=1$ (survival)

otherwise $C_{9,n+1}=0$ (death)

if $C_{9,n}=0$ and $\sum_{r=1}^{8} C_{r,n}=3$ then $C_{9,n+1}=1$ (birth)

otherwise $C_{9,n+1}=0$. (quiescence)

CONWAY'S RULES OF 'LIFE' FIGURE C2



EXAMPLES OF 'LIFE' PRODUCING SYMMETRY FIGURE C3 of Von Neumann and Codd, would it not be possible to make a closer analogy to nature by creating similar memory chains in "cellular state space"; that is, allow for a bifurcation transition state that could exploit the reflexion symmetry of the transition space and still perform primary functions. In both these examples, it has been the transition matrix that had had symmetric properties and these have been reflected in the output. It is therefore felt that a study of symmetric transition matrices should be made since they reflect the inherent capabilities of a finite state machine.

Of particular interest is John Horton Conway's 'game of life', which is played out on a regular memory matrix (such as a go-board with counters, or a VDU display) and has the following carefully thought-out rules:

1 Survivals. Every counter with two or three neighbouring counters survives for the next generation.
2 Deaths. Each counter with four or more neighbours dies (is removed) from over population. Every counter with one neighbour or none dies from isolation.
3 Births. Each empty cell adjacent to exactly three neighbours - no more, no fewer - is a birth cell. A counter is placed on it at the next move.

which are summarised in Figure C2. On playing the game many fascinating shapes are generated from initially quite simple population (Martin Gardner 1970). One of the most striking is that, to quote Gardner verbatim:

> "Patterns with no initial symmetry tend to become symmetrical. Once this happens the symmetry cannot be lost, although it may increase in richness".

(See Figure C3).

Thus it appears that embedded in this complex production is a way of generating symmetry and sustaining it. However, it should be noted that there is no noise (ie no uncertainty) in Conway's game of Life. Study in this area must continue.

Lastly it should be re-emphasised that symmetry is best thought of as a sort of equilibrium - a balancing of scales such that stability is ensured, a perfect concept matched to an imperfect universe where snowflakes melt, galaxies form and tigers are shot.

REFERENCES

ASHBY, W. ROSS, "An Introduction to Cybernetics", Chapman and Hall, 1956, p206

BEER, STAFFORD, "Platform for Change", John Wiley and Sons, 1975, pp110-111

BEVERIDGE, G.S.G. AND SCHECHTER, R.S., "Optimisation: Theory and Practice", McGraw-Hill Kogakusha, 1970, pp368-383

BRONOWSKI, J., "The Ascent of Man", BBC Publications, 1973, pp160-172

BROWN, B.M., "The Mathematical Theory of Linear Systems", Chapman and Hall, 1961, pp119-122

BROWN, J. AND GLAZIER, E.V.D., "Telecommunications", Chapman and Hall, 1964, p206

CHERRY, C., "On Human Communication", MIT Press, 1957, pp269-271

EYKHOFF, P., "System Identification: Parameter and State Estimation", Wiley, 1974, p167

GARDNER, M., "Mathematical Games: The Fantastic Combinations of John Conway's New Solitaire Game 'Life'", Scientific American, November 1970, pp120-123

HAYES, D., (1) "A Cybernetic Approach to Pattern Recognition", M.Phil. Thesis, Chelsea College, University of London, 1976, pp93-114

HAYES, D., (2) "A Technique for Automatically Identifying the Axis of Symmetry of Cophasal Array Antenna Patterns", RADAR 77, IEE Conference, 1977, pp323-327

HUBEL, D.H. AND WIESEL, T.N., "Functional Architecture of Macaque Monkey Visual Cortex", Ferrier Lecture 1972, Proceedings of the Royal Society of London, Biological Sciences. Volume 198, 28 July 1977, pp41-43

MINSKY, M., "Computation: Finite and Infinite Machines", Prentice Hall International Inc., London, 1967, p228

MINSKY, M. AND PAPERT, S., "Perceptrons: An Introduction to Computational Geometry", The MIT Press, 1969, pp117-120

ULLMANN, J.R., "Pattern Recognition Techniques", Butterworth, 1973, p117

WEYL, H., "Symmetry", Princeton University Press, 1952, pp26-38

WIENER, N., (1) "Cybernetics or Control and Communication in the Animal and the Machine", MIT Press, 1948, pp12-13

WIENER, N., (2) "The Human Use of Human Beings", Sphere Library, 1968 - first published USA 1950, p17

APPENDIX ONE

A PAPER ON THE APPLICATION OF SYMMETRIC SIGNAL DETECTION THEORY TO A RADAR SYSTEM

Foreword

The following paper was presented by the author at RADAR 77 and describes a method for detecting symmetric signals that occur when a scanning antenna (of symmetric pattern) receives a constant signal at fixed azimuth.

It also details the form of the simulation of the system threshold optimisation techniques and gives some results relating to the probabilities of detection and probabilities of false alarm.

A TECHNIQUE FOR AUTOMATICALLY IDENTIFYING THE AXIS OF SYMMETRY OF CO-PHASAL ARRAY ANTENNA PATTERNS

ABSTRACT

A theory of symmetric signal detection is developed and subsequently applied to computer-simulated returns expected from an hypothetical, electronically scanned, circular array, surveillance radar.

The performance of the symmetry detection process is assessed by accumulating the probabilities of detection and of false alarm for single signals within a multiple signal environment, from these measurements it is possible to define an average measure of the detection efficiency.

This measure is used to act as an objective function for a sequential simplex optimisation scheme; which is in turn used to improve the threshold values needed in the detection process.

DAVID HAYES

Marconi-Elliott Avionic Systems Limited.

A TECHNIQUE FOR AUTOMATICALLY IDENTIFYING THE AXIS OF SYMMETRY OF CO-PHASAL ARRAY ANTENNA PATTERNS

D. Hayes

Marconi-Elliott Avionic Systems Limited

INTRODUCTION

Since the early 1960's a considerable degree of effort has been directed towards the study and development of electronically scanned, circular array, antenna systems - as is indicated by the bibliography given by Wardrop (1). When such a configuration is used with an omni-directional source, as was proposed by McCartney (2), the associated returns are locally symmetric functions that correspond closely to the directional pattern of the array, details of which are given in a paper by Fenby and Davies (3), together with a photograph of the actual returns. In fact the axes of local symmetry of the returns lie at the relative bearings of the illuminated targets. Clearly, if within-pulse scanning is to be used it is essential the axes of local symmetry be detected as precisely as is possible, in order that the azimuth information may be accurately displayed and processed.

SYMMETRIC SIGNAL DETECTION

The conventional detection process for symmetric signals with reduced sidelobes is to record the time instants when a preset threshold is crossed over and then to take the average of these two instants to deduce the centre of the signal and hence the bearing of the beam in the case of a circular array. This method has the advantage of simplicity, but plainly is capable of generating false alarms (due to sidelobe detection) and failures to detect (due to signals being below threshold) when there is a wide range of signal levels to contend with. Other methods can be envisaged which involve the use of cross and auto-correlation techniques, but these are more complex and demand substantial storage and computation. The process to be described is a "midway house" which exploits the symmetry of the situation.

The Symmetry Metric

The simplest formulation of the symmetry metric $S(\theta)$ is for the case of continuous returns $R(\theta)$ and is of the form:

$$S(\theta) = \int_{\alpha=0}^{\alpha=\epsilon} |R(\theta+\alpha) - R(\theta-\alpha)|\, d\alpha \ldots\ldots\ldots\ldots(1)$$

where θ is the beam bearing, α is a dummy variable and ϵ is the extent of the returns - 2ϵ is the window length.* The above integral may be interpreted as performing the operation of folding the returns back on themselves about a beam bearing θ and accumulating the moduli of their differences over a window 2ϵ, to produce a measure of the error in (or deviation from) symmetry. It appears to embody the virtues of mask-matching and auto-correlation, in that a differencing procedure is adopted which corresponds closely to simple mask-matching (not involving multiplication), while at the same time avoiding the need to store a set of reference patterns - as does auto-correlation.

In practice it would seem likely that R (θ) would be regularly sampled and therefore expression (1) will now be reformulated to meet this contingency. If α, θ and ϵ are now read as discrete variables, the symmetry metric about θ may be defined in two ways:

$$S_{even}(\theta) = \sum_{\alpha=0}^{\alpha=\epsilon-1} |R(\theta+\alpha+1) - R(\theta-\alpha)| \ldots\ldots(2a)$$

and

$$S_{odd}(\theta) = \sum_{\alpha=1}^{\alpha=\epsilon} |R(\theta+\alpha) - R(\theta-\alpha)| \ldots\ldots\ldots(2b)$$

this is because the sampling process can give rise to two perfect symmetries - one arising out of an even number of samples ($S_{even}=0$) the other out of an odd number ($S_{odd}=0$). Normally, the sampled signal will tend more closely to one or the other of these two discrete symmetries and $S(\theta)$ may therefore be redefined as:

$$S(\theta) = \min\{S_{even}(\theta), S_{odd}(\theta)\} \ldots\ldots\ldots\ldots(2)$$

for low sampling rates. The difference between S_{even} and S_{odd} will of course be marginal for high sampling rates and hence the most convenient may be chosen under these circumstances.

The Discriminant

In order that the decision making process may be completed a set of threshold discriminants $\mathcal{D}$ may be applied to $S(\theta)$ and some joint decision reached as to the signal's status at a bearing θ. Thus if, in general:

$$\mathcal{D} = \{D_1, D_2 \ldots D_r \ldots D_n\} \ldots\ldots\ldots\ldots\ldots(3)$$

and if, for instance:

$$D_r(\theta) = \lceil S(\theta+r) > S(\theta)+\rho_r \rceil \wedge \lceil S(\theta-r) > S(\theta)+\rho_r \rceil \ldots(4)$$

where ρ_r is the rth threshold and the operational brackets "$\lceil\ \rceil$" assign the enclosed expression a value of '0' or '1' according as to whether or not the enclosed expression is true (=1) or false (=0). A final joint decision $D(\theta)$ as to a symmetric signal's presense at a bearing θ, may, for example, be formulated as:

$$D(\theta) = \lceil \sum_{r=1}^{r=n} D_r(\theta) \geq \rho \rceil \ldots\ldots\ldots\ldots\ldots\ldots(5)$$

*n.b. Relative angles (α, θ, ϵ) rather than times (t, T, τ) have been used in this text as they are thought more appropriate and easier to visualise, but obviously transposition is possible, provided causality is satisfied in the initial definition of T and the system employs linear scanning.

where ρ is a threshold, such that $0<\rho\leq n$; then an unanimous decision will be reached if $\rho=n$, and a majority decision if $\rho = n/2+1$, in evaluating $D(\theta)$. Expression (4) is only one of many possible predicates, but does perform the essential task of testing for a sharp minimum when incorporated into $D(\theta)$, which identifies the overall qualities of $S(\theta)$ and thus, indirectly, $R(\theta)$.

A COMPUTER SIMULATION

The computer simulation shown in Figure 1 has been used by the author to assist in the design of an optimum system for the detection of locally symmetric signals. The simulation can be seen to be divided into four main areas (excluding the control unit and the graphical display) which will now be discussed.

The Data Synthesiser

The data synthesiser provides a simulated environment in which the signal processing can be tested. In this initial study it has been thought unnecessary to model the radar's environment in absolute detail, but only to retain the primary aspects of the returns; which are that:

a. they are regularly sampled,
b. they approximate to the beam pattern,
c. they occur randomly,
d. they are subject to magnitude fluctuations

and

e. they have a background of white noise.

Plainly, more sophisticated models, which include such facets as clutter, quantisation and edge effects, are possible, but perhaps not wholly desirable in this pilot investigation.

In Figure 1 the data synthesiser is sub-divided into four procedures which interact to produce a sampled return as is displayed in Figure 2a - interaction between adjacent patterns can be seen to cause asymmetries that make subsequent analysis difficult.

The Signal Processor

The signal processor is split into two sub-units: a symmetry detector and a discriminator, the general operation of which has already been described.

The simulated returns are operated on by expression (2) to produce a transformation as is shown in Figure 2b by the continuous line graph, in order to distingguish it from the returns. By changing the value of the extent δ, more or less of the overall pattern may be taken into account and by changing the standard deviation of noise and the sidelobe level their effects on the symmetry metric can be monitored.

The discriminant that was mainly used in the simulation was:

$$D(\theta) = \lceil S(\theta-2)>S(\theta) + \rho_1 \rceil \wedge \lceil S(\theta+2)>S(\theta)+\rho_1 \rceil \wedge \lceil S(\theta-1) > S(\theta) + \rho_2 \rceil \wedge \lceil S(\theta-1)>S(\theta)+\rho_2 \rceil \wedge \lceil R(\theta)>\rho_3 \rceil \quad \ldots(6)$$

which requires an unanimous decision to be made as to a signal's authenticity. In order to avoid spurious detections at points between adjacent signals of the same magnitude a further test has been incorporated that determines that the returns are indeed above a certain minimum level.

The effectiveness of expression (6) can be assessed initially by studying displays of the type shown in Figure 2c, where the true signal locations are indicated as upward pointing arrows and the detected locations by downward pointing arrows. As has been mentioned analysis becomes difficult when the patterns merge and hence the false identification shown. The values of ρ_1, ρ_2 and ρ_3 can as a first attempt be read from graphs of the type shown in Figure 2b.

The Statistical Analyser

The statistical analyser accurately assesses the performance of the signal processor by calculating the probabilities of detection 'P_d' and of false alarm 'P_{fa}' by averaging the associated single scan probabilities 'p_{fa}' and 'p_d'. Thus if a is the number of returns identified by the signal processor and i is the number of signals generated:

$$p_d(i) = 1 - \text{number of signals not detected}/i$$

and

$$p_{fa}(i) = \text{number of signals falsely detected}/a.$$

The two numerators may be calculated, since the generated locations and the detected locations are known for the Tth scan - that is, if I and J are arrays storing these locations one may simultaneously let:

$$I(T,r) := I(T,r).\lceil |I(T,r)-J(T,t)| > s \rceil \text{ for all } t \,\&\, r \ldots(7)$$

and

$$J(T,t) := J(T,t).\lceil |I(T,r)-J(T,t)| > s \rceil \text{ for all } t \,\&\, r \ldots(8)$$

where ":=" is read as 'takes the value', s is a positive integer tolerance and t and r are dummy variables, such that $1\leq t\leq a$ and $1\leq r\leq i$. Thus I and J are left unchanged if there is no correspondance and set equal to zero if there is complete correspondance, so that all that remains in these arrays are the signal locations that do not correlate.

Hence from these re-assigned arrays the probabilities p_d and p_{fa} may be calculated, since:

$$p_d(T,i) = 1 - 1/i \sum_{r=1}^{r=i} \lceil I(T,r) \neq 0 \rceil \ldots\ldots\ldots\ldots(9)$$

and

$$p_{fa}(T,i) = \sum_{t=1}^{t=a(T)} \lceil J(T,t) \neq 0 \rceil / a(T) \ldots\ldots\ldots\ldots(10)$$

where a(T) is the number of signals detected on the Tth scan. Expressions (9) and (10) may then be averaged to give:

$$P_d(i) \simeq 1/N \sum_{T=1}^{T=N} p_d(T,i) \ldots\ldots\ldots\ldots(11)$$

and

$$P_{fa}(i) \simeq 1/N \sum_{T=1}^{T=N} p_{fa}(T,i) \ldots\ldots\ldots\ldots(12)$$

The value of N may be determined by incorporating a percentage accuracy test in this area of the simulation.

The results of this section are summarised by the four graphs shown in Figure 3, the behaviour of which are typical and can be explained in terms of signal interaction causing false alarms and occlusions.

The System Optimiser

The system optimiser is intended to improve the overall performance of the detection process by automatically optimising the designer's measurements of the thresholds used by the discriminator. This is only possible because both the synthesised and the identified bearings are available and can be used by the statistical analyser to produce the probabilities of detection and of false alarm, on which an estimate of the system efficiency can be based for any set of thresholds/parameters. Thus, a structured adjustment of the thresholds is feasible until a maximum efficiency is arrived at.

Before this adjustment can be executed an objective function must be decided upon. One measure of efficiency 'E' that might serve as an objective function is:

$$E(\rho_1\ \rho_2\ \rho_3) = {}^1/_m \sum_{i=1}^{i=m} (P_d\,(i)+(1-P_{fa}\,(i)))/_2 \ldots\ldots\ldots(13)$$

where m is the maximum number of signals considered and E an average of the probabilities of detection and of no false alarm for a range of signals that ideally should have the value one and in any event should be maximal for some set of thresholds - $\{\rho_1\ \rho_2\ \rho_3\}$.

An optimisation routine known as: "the sequential simplex method" has been used, which is in brief based, in the three dimensional case, on the evaluation of the objective function at the four vertices of an initially, arbitrarily orientated, tetrahedron which is intelligently placed in the threshold space. This tetrahedron, which in n-space is called "a simplex", is made to search through the threshold space by following the algorithm that its minimum vertex should always be replaced by its mirror image which will hopefully have an higher value of threshold associated with it. This process continues, with certain precautions to prevent oscillation on ridges, until a fixed point is cycled round, when either the simplex is reduced in size or the maximum is said to have been found within the tolerance of the simplex's radius. For a more detailed explanation of this technique the reader is referred to Spendley et al (4).

The final results of the optimisation will vary according to the actual external parameters chosen (e.g. the sampling rate, the sidelobe level etc.,) and the type of objective function used. Similar methods can, of course, be applied to the design of the directional pattern, which may have constraints imposed. For example, the number of elements in a circular array is restricted to an integer value and will for mechanical/electronic considerations be constrained to a certain maximum number. Under these circumstances an optimal balance between the directional pattern and the detection process can be sought by using a variation on the sequential simplex theme, known as the "complex method of Box", which encompasses the optimisation of both integer and real parameters - with or without constraining inequalities - and is well described by Beveridge and Schechter (5).

CONCLUSIONS

The theory and simulations just outlined have led the author to believe that symmetric signal detection is, in principle, viable and has certain advantages over conventional techniques in that it requires little storage, a minimum of computation and will detect the axis of symmetry of a multimodal pattern, irrespective of sidelobe level, provided it is symmetric - this may be significant in the case of within-pulse scanning where a "beam-split" may occur due to errors in phase compensation (3).

However there are some difficulties to be overcome in the application of symmetric signal detection to a circular scanned array. One problem is that for a fixed range cell the return may begin anywhere in the cell and start at any point in the pattern - thus causing the extent of local symmetry to fluctuate. To avoid this problem several lines of action appear to present themselves, for example the scanning rate may be increased to two scans per illuminating pulse so that the directional pattern is seen at least once per range cell without edge effects or, alternatively, it might be possible to adopt a scheme whereby the beginnings and ends of successively overlapping range cells are looped around on themselves by a modulo arithmetic scheme applied to the angular measurement.

Clearly much interesting work remains to be done in the form of comparative studies of the accepted and proposed detection methods; this will involve further sophistications in processing and modelling, in order to assess fully the value of the new technique that changes the emphasis from the main beam-to-sidelobe level ratio to that of the symmetry of the directional pattern.

REFERENCES

1. Wardrop, B : 'A Synthesis Procedure for Circular Arrays'. The Marconi Review, 1975, Fourth Quarter, p. 184.

2. McCartney, B.S. : 'Proposals for an Electronically Scanned Circular Array'. Proc. I.E.E., 1963, 110, No. 7, p. 1220.

3. Fenby, R.G. and Davies D.E.N. : 'Circular Array providing Fast 360° Electronic Beam Rotation'. Proc. I.E.E., 1968, 115, No. 1, p. 78.

4. Spendley, W., Hext, G.R. and Himsworth, F.R. : 'Sequential Applications of Simplex Design in Optimisation and Evolutionary Operation'. Technometrics, 1962, No. 4, p. 441.

5. Beveridge, G.S.G and Schechter, R.S. 'Optimisation, Theory and Practice'. McGraw Hill, 1970.

ACKNOWLEDGEMENT

The author wishes to acknowledge the support of the Procurement Executive, Ministry of Defence, in the carrying out of this work.

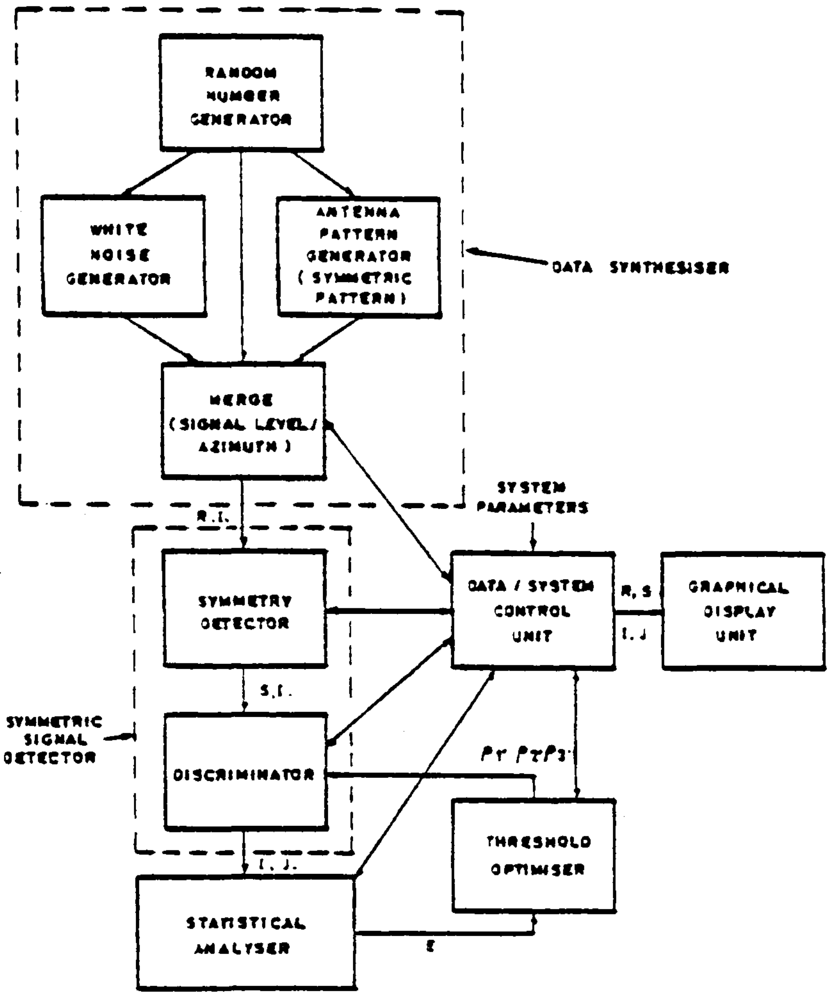


Figure 1 Computer Assisted Design System

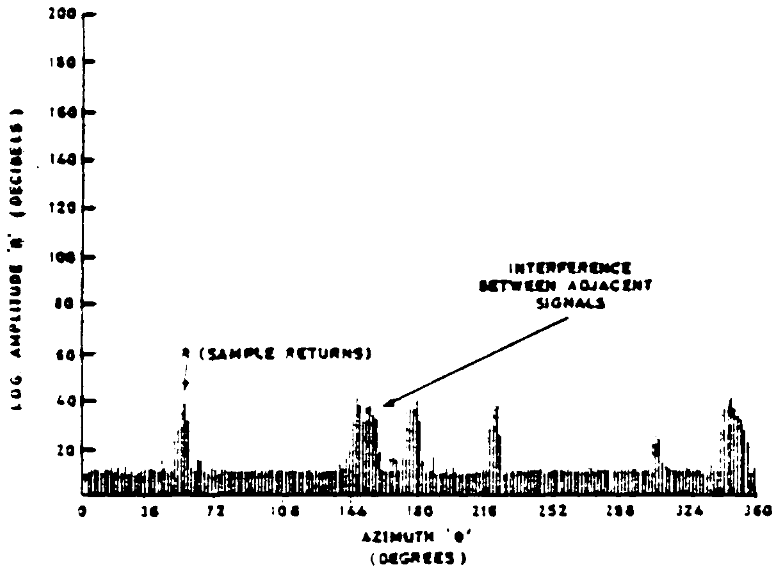


Figure 2a Synthesised Returns

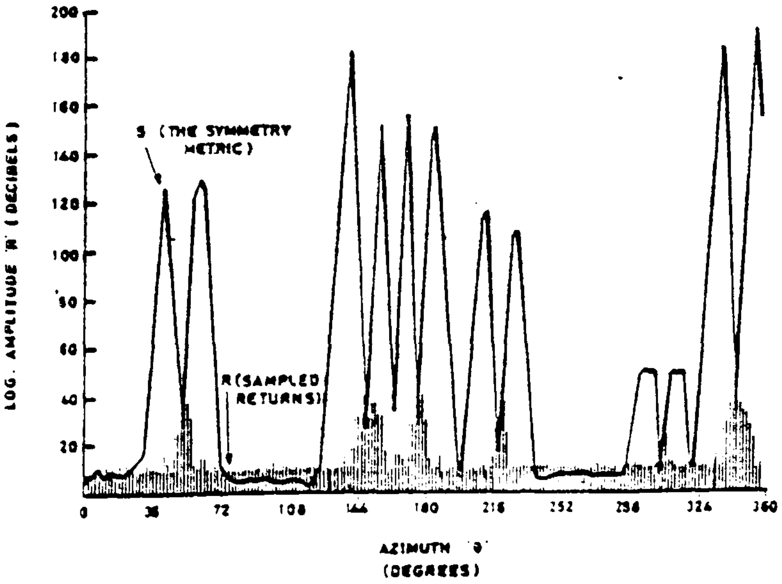


Figure 2b Symmetry Metric

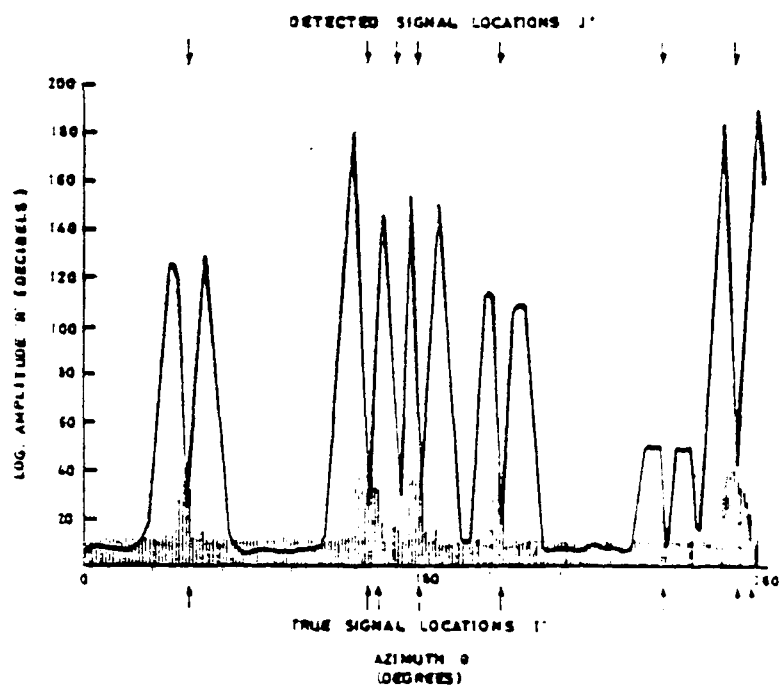


Figure 2c Detection Process

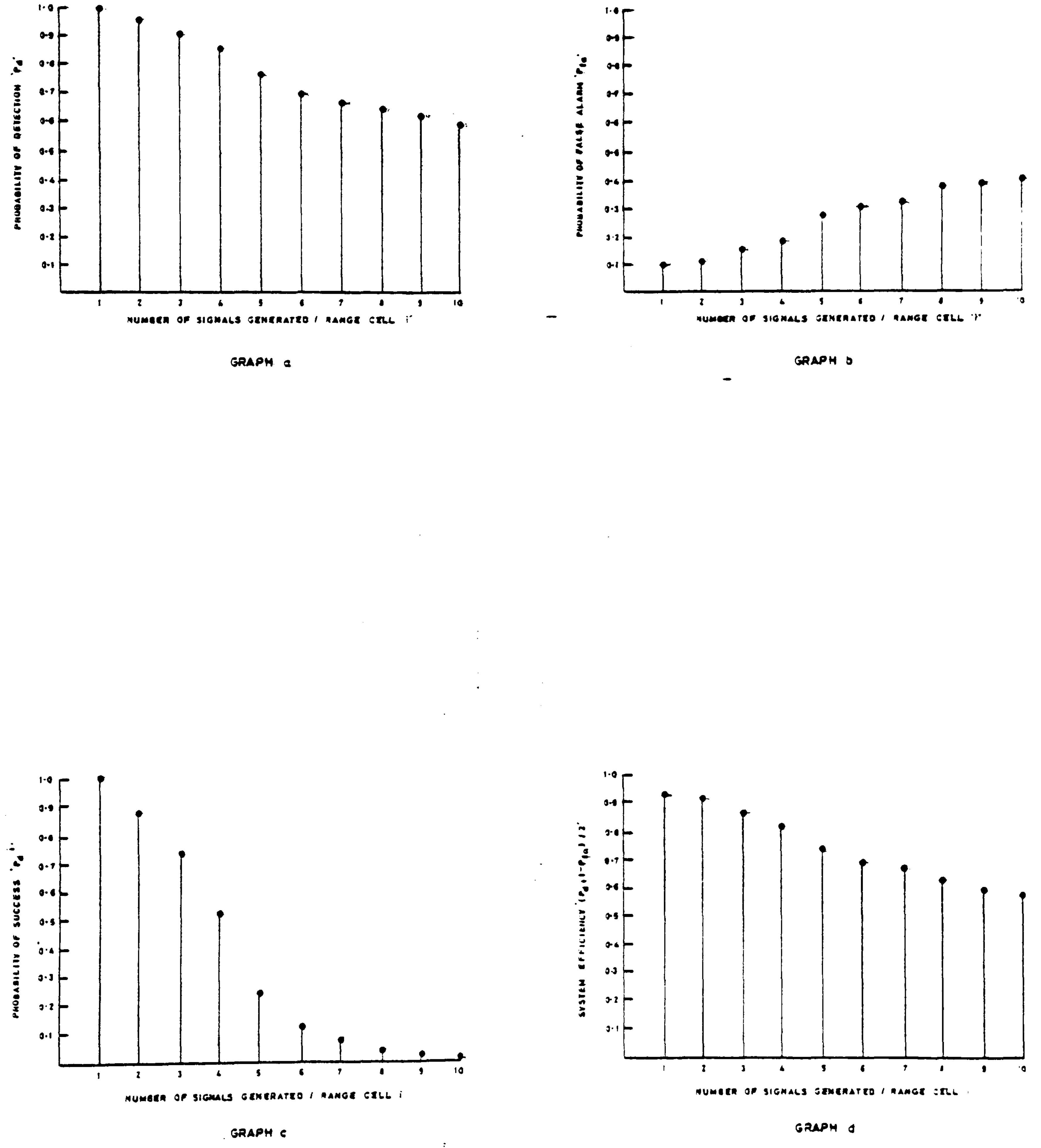


Figure 3 Graphical Representations of System Performance

APPENDIX TWO

A COMPUTER PROGRAM FOR THE SIMULATION AND OPTIMISATION OF A RADAR SYSTEM

Foreword

The following computer program was written in ALGOL for the Elliott 503 computer (apart from a random number generator in Elliott machine code). The program performs the tasks described in Appendix One for simulating a radar system.

Also included are several graph plotting procedures that can be called by a program executive at the end of the program which controls the program routing and is where all necessary input parameters are read in.

It should be understood that there has been no effort made to optimise the code.

```
comment  COMPLETE SIMULATION WITH SIMPLEX OPTIMISATION SCHEME;
begin integer i, n1,n2,n3,n4,a,b,d,p,m,g,k,n,s,r,t,M1,s1,N,M,sw1,
sw2,sw3,sw4;
real f,c,h,l,MM,u,v,w,e,e1,e2;
array A1,B,C,D,E,P1,P2,P3,P4[0:15], x[1:3], I,J[0:100], A[-260:460],
P[-200:200], mean,phim,psim[-60:260];

procedure plot 1 (W,x,y,d);

comment    This procedure plots the data from an array W
           as a continuous graph in the first quadrant -
           the maximum positive x and y values must be
           supplied, together with a value of 1 or 0 for
           d, according to whether or not axes are required;

value  x,y,d;
integer  x,y,d;  array W;
begin integer z;
setorigin (300,400/x,300/y,1);
if d=1 then axes (x/10,y/10,10,0,10,0);
for z:=1 step 1 until x do begin
if z=1 then movepen(z,W[z]);
if z>1 then drawline (z,W[z]); end; end;

procedure plot 2 (W,x,y,d);

comment    This procedure plots the data from an array W
           as a line graph in the first quadrant -
           the maximum positive x and y values must be
           supplied, together with a value of 1 or 0 for
           d, according to whether or not axes are required;

value  x,y,d,
integer  x,y,d;  array W;
begin integer z;
setorigin (300,400/x,300/y,1);
if d=1 then axes (x/10,y/10,10,0,10,0);
for z:=1 step 1 until x do begin
if z>1 then movepen(z,0);
if z>1 then drawline (z,W[z]); end; end;

procedure plot 3 (I,n,x,y,d);

comment    This procedure plots upward pointing arrows
           underneath the horizontal axis.  A switch d,
           of value 0 or 1, is included in order to control
           the drawing of axes, an array I provides the
           x coordinate of the arrow;

value n, x,y,d;
integer  n,x,y,d;  array I;
begin integer z;
setorigin (300,400/x,300/y,1);
if d=1 then axes (x/10,y/10,10,0,10,0);
for z:=1 step 1 until n do begin
movepen(I[z],-10); concharacter (10);
end; end;
```

```
procedure plot 4 (I,n,x,y,d);

comment    This procedure plots downward pointing arrows
           above the horizontal axis.  A switch d,
           of value 0 or 1, is included in order to control
           the drawing of axes, an array I provides the
           x coordinate of the arrow;

value  n,x,y,d;
integer  x,n,y,d;  array I;
begin integer z;
setorigin (300,400/x,300/y,1);
if d=1 then axes (x/10,y/10,10,0,10,0);
for z:=1 step 1 until n do begin
movepen(I[z],y+10); cencharacter (8);
end; end;p

real procedure rand;

comment    This procedure generates a random number,
           lying between -1 and 1 and of rectangular
           distribution;

begin real x;
switch ss:=L;
elliott (3,0,n1, ,5,2,n2);
elliott (5,4,20,0,1,0,n1);
elliott (2,0,n2,0,4,3,L);
L:  x:=n2/274877906994.0;
if abs(x)=0 then
begin print punch (3), £zero random number?;
wait;
end;
rand:=x;
end rand;

real procedure noise;

comment    This procedure generates truncated Gaussian
           noise - by use of the random number generator;

begin real p;
integer k;
p:=0;
for k:=1 step 1 until 12 do p:=p+rand;
noise:=p/2;
end noise;
```

```
procedure aerial pattern;

comment     This procedure generates antenna patterns
            which have been regularly sampled;

begin real pi,w,x,y,z,rand1;
integer j;
pi:=4*arctan(1);
begin y:=rand*e2*pi;
rand1:=(rand+1)/2;
rand1:=c-(c-1)*rand1;
for j:=-g step 1 until g do
begin  x:=j*2*pi/8+2*pi*rand/16;
z:=(x*x+y*y)↑0.5;
if z<.000001 then w:=1 else
w:=(abs(sin(z)/z))↑h;
P[j]:=rand1*w;
end;
end;
end aerial pattern;

procedure merge;

comment     This procedure merges together the antenna
            patterns and the random noise in one azimuth
            array;

begin integer j,n,k;
switch z:=ZERO;
for k:=-260 step 1 until 460 do
A[k]:=0;
if i=0 then goto ZERO;
for n:=1 step 1 until i do
begin
j:=entier(100*rand+99);
I[n]:=j,
aerial pattern;
for k:=-g step 1 until g do
A[j+k]:= P[k] + A[j+k];
end;
ZERO: for k:=-40 step 1 until 240 do begin
if A[k]≠0 then A[k]:=10*ln(A[k]/10+(noise+6)*f/200)/ln(10) + 1;
if A[k]<0 then A[k]:=0;
A[k]:=A[k] + MN; end;
end merge;
```

```
procedure prerecognition;

comment   This procedure resets the initial conditions;

begin    i:=0;   n1:=51772746+n3;   n2:=31 41867+n4;   end;

procedure recognition;

comment   This procedure identifies the centres of
          possible antenna patterns and records the
          results;

begin  merge;
for n:=-40 step 1 until 240 do
phim[n]:=psim[n]:=mean[n]:=0;
for n:=-20 step 1until 220 do
begin for s:=-k step 1 until k do
mean[n]:=A[n+s]+mean[n];
mean[n]:=mean[n]/(2*k+1);
end;
a:=0;
for n:=-10 step 1 until 210 do begin
for s:=0 step 1 until m do begin
phim[n]:=abs(mean[n-s] - mean[n+s+1])+phim[n];
psim[n]:=abs(mean[n-s] - mean[n+s])+psim[n];
end;
if phim[n]<psim[n] then psim[n]:=phim[n];
end;
for n:=0 step 1 until 200 do begin
if psim[n-1]>(psim[n]+u) and psim[n-2]>(psim[n]+v)
and psim[n+1]>(psim[n]+u) and psim[n+2]>(psim[n]+v)
and mean[n]>(MN+w)
then  begin a:=a+1; J[a]:=n;. end;
end;  end;

procedure analysis;

comment   This procedure statistically analyses the
          performance of the recognition system
          and accumulates the probabilities of false
          alarm and of success - the average of which
          is used in the simplex procedure as an
          objective function;

begin switch S:= ENTER, RESETt, RESETr, RETURN;
RETURN :          recognition;
```

```
if a=0 then a:=1;
t:=1;
RESETt: r:=1;
RESETr:  if abs(J[t]-I[r])<s1 then begin
J[t]:=0;    I[r]:=0;
goto ENTER;
end
else r:=r+1;
if r<i then  goto RESETr;
ENTER:  t:=t+1;
if t<a then goto RESETt;
for r:=1 step 1 until i do
if I[r]>0.1 then D[i]:=D[i]+1;
for t:=1 step 1 until a do
if J[t]>0.10 then E[i]:=E[i]+1;
B[i]:=E[i]/a+B[i];   E[i]:=0;
i:=i+1;
if  i<d then goto RETURN;
M:=M+1;
n3:=n3+1;   n4:=n4+1;
if M<M1 then begin prerecognition;  goto RETURN end;
e:=0;
for i:=1 step 1 until d  do
begin
P1[i]:=B[i]/M1;  P2[i]:=D[i]/(i*M1);  P3[i]:=1-(P1[i]+P2[i])/2;
P2[i]:=1-P2[i];  P4[i]:=P2[i]↑i;  e:=P3[i]+e; end;

for i:=1 step 1 until 15 do
B[i]:=D[i]:=0;
for i:=1 step 1 until 100 do
J[i]:=I[i]:=0;  M:=1; i:=0;     end;

real procedure objective function (P,r);

comment    This procedure acts as an interface between
           the simplex procedure and the rest of the
           program;

value r;   integer r;    array P;

begin analysis;
u:=P[1,r];  v:=P[2,r];   w:=P[3,r];
objective function:=e;
end;
```

```
procedure simplex(x,n,a,b,s);

comment    This procedure optimises a system of n real
           parameters, by evaluating an objective function
           (which must be supplied as a real procedure
           in the main part of the program) at n+1 points,
           a distance a units apart - a distribtion known
           as a simplex.  The simplex is sequentially
           adjusted to search through the n-space parameters
           until an optimum in the objective function -
           to within a percentage tolerance b - is found.
           The objective function is assumed to be unimodal;

value n, a, b, s;  real a, b;  integer n, s;  array x;

begin  real p,q,r,ymin,ymax,ymaxm;  integer h,i,j,k,l,m;
array X[1:n,1:n+1], Y[1:n+1];
switch s1:= CHECK, START, FINISH;

comment    Before the search progresses the initial simplex
           must be generated;

r:=0;
START:   p:=a/n/(2↑0.5)*((n+1)↑0.5+n-1);  q:=a/n/(2↑0.5)*((n+1)↑0.5-1);

for j:=1 step 1 until (n+1) do
for i:=1 step 1 until n do begin
if i=(j-1) then X[i,j]:=x[i]+p else X[i,j]:=x[i]+q;
if j=1 then X[i,j]:=x[i]; end;

comment    Next the objective function must be evaluated
           at points on the initial simplex;

ymin:=1⏨60;  ymax:=ymaxm:=-ymin;  l:=-1; m:=0;

for j:=1 step 1 until (n+1) do
Y[j]:=objective function(X,j);

if s=1 then begin
for j:=1 step 1 until (n+1) do begin print punch (3), Y[j], ££s??;
for i:=1 step 1 until n do print punch (3), sameline, X[i,j]; end;  end;

comment    This is followed by the checking for a
           minimum vertex and the subsequent adjustment
           of the simplex;

CHECK:  for j:=1 step 1 until (n+1) do begin
if Y[j]<ymin and j≠1 then begin ymin:=Y[j]; k:=j; end;
if Y[j]>ymax then begin  ymax:=Y[j]; h:=j; end;  end;
for i:=1 step 1 until n do begin p:=0;
for j:=1 step 1 until (n+1) do p:=X[i,j]+p;
X[i,k]:=2/n*(p-X[i,k])-X[i,k]; end;
```

```
Y[k]:=objective function(X,k);

if s=1 then begin
print punch(3),Y[k], ££s??;
for i:=1 step 1 until n do print punch (3), sameline, X[i,k];
print punch(3), sameline, m; end;

comment    In order to terminate the procedure, a
           test to determine the stability of the maximum
           point in the simplex must be incorporated;

if 1₁₀-60>abs(ymaxm-ymax) then m:=m+1 else m:=0;
if m>(1.65*n+0.05*n*n) then goto FINISH;
l:=k;  ymin:=1₁₀60;  ymaxm:=ymax;
goto CHECK;

FINISH:  for i:=1 step 1 until n do x[i]:=X[i,h];
if abs((Y[h]-r)/Y[h]*100))>b then begin
print punch (3), ££l??,  £the reduced simplex size is?, sameline, a/1 , ££l??;
 a:=a/10; r:=Y[h]; goto START; end;

print punch (3), ££l??, £ THE OPTIMUM VALUES ARE ?, ££l??, Y[h], ££l??;
for i:=1 step 1 until n do print punch (3), X[i,h];
end;



comment    This is the main program, which controls all
           the former procedures, as well as providing the
           initial conditions for the system to operate;

begin  switch ss:=S1,S2,S3;
for n:=0 step 1 until 15 do
A[n]:=B[n]:=C[n]:=D[n]:=E[n]:=P1[n]:=P2[n]:=P3[n]:=P4[n]:=0;
for n:=0 step 1 until 100 do
J[n]:=I[n]:=0; M:=1;
print punch (3), £Select one of the following programs by typing?,
££l??, £the number associated with it?, ££l??,
£ (1) Signal Power/Azimuth?, ££l??,
£ (2) Probabilities of Detection and of False Alarm?, ££l??,
£ (3) System Optimisation?, ££l??;
read reader (3), sw1;
if sw1=1 then begin print punch (3),
£ Select one of the following display formats by typing ?,
££l??, £ the number associated with it?, ££l??,
£ (1) A line graph of noise/azimuth?, ££l??,
£ (2) A continuous graph of noise/azimuth?, ££l??,
£ (3) As (1) but with symmetry function superimposed as a continuous graph?, £
```

```
££1??,
£ (4) As (1) but with recognition arrows?, ££1??,
£ (5) As (3) but with recognition arrows?, ££1??,
£ (6) As (3) but without lower arrows?;
S1: read reader (3), sw2;
print punch (3), £ The following data are required - ?, ££1??,
£ i,  f,  MN,  n3,   n4,  k,  m,  g,  h,  l,  e2,  c,  p,  u,  v,  w?;
read reader (3), i,f,MN,n3,n4,k,m,g,h,l,e2,c,p,u,v,w;
n1:=n3+51772746;  n2:=n4+31041867;
recognition;
if sw2=1 then
plot 3 (A,200,p,1);
if sw2=2 then
plot 1 (A,200,p,1);
if  sw2=3 then begin
plot 2 (A,200,p,1);
movepen (0,0);
plot 1 (psim,200,p,0);end;
if sw2=4 then begin
plot 2 (A,200,p,1);
movepen (0,0);
plot 3 (I,i,200,p,0);
movepen (0,0);
plot 4 (J,a,200,p,0);  end;
if sw2=5 then begin
plot 2 (A,200,p,1);
movepen (0,0);
plot 1 (psim,200,p,0);
movepen (0,0);
plot 3 (I,i,200,p,0);
movepen (0,0);
plot 4 (J,a,200,p,0);end;
if sw2=6 then begin
plot 2 (A,200,p,1);
movepen (0,0);
plot 1 (psim,200,p,0);
movepen (0,0);
plot 4 (J,a,200,p,0); end;
goto S1  end;
S2:  if sw1=2 then begin
print punch (3), £The following data are required?, ££1??,
£ f,  MN,  n3,  n4,  k,  m,  g,  h,  l,  e2,  c,  d,  M1,  s1,  u,  v,  w ?;
read reader (3), f,MN,n3,n4,k,m,g,h,l,e2,c,d,M1,s1,u,v,w;
prerecognition;
analysis;
plot 2 (P1,d,1,1);
movepen (0,3/2),
plot 2 (P2,d,1,1);
movepen (0,3/2);
```

```
plot 2 (P3,d,1,1);
movepen (0,3/2);
plot 2 (P4,d,1,1);
goto S2;  end;
S3:  if sw1=3 then begin
print punch (3), £ The following data are required-?, ££1??,
£ f, MN,  n3,  n4,  k,  m,  g,  h,  l,  e2,  c,  d,
   M1,  s1,  u,  v,  w,  a,  b,  s?, ££1??;
read reader (3),  f,MN,n3,n4,k,m,g,h,l,e2,c,d,M1,s1,x[1],x[2],x[3],a,b,s;
prerecognition;
simplex (x,3,a,b,s);
plot 2 (P1,d,1,1);
movepen (0,3/2);
plot 2 (P2,d,1,1);
movepen (0,3/2);
plot 2 (P3,d,1,1);
movepen (0,3/2);
plot 2 (P4,d,1,1);
goto S3;  end;

end;  end;  end;  end;
```

APPENDIX THREE

A GENERAL PAPER ON THE CYBERNETICS OF SEEING

Foreword

The following paper is a summary of work of the author over the last 6 years. It includes a mathematical explanation of vector chain mapping in relation to contour following and covers such symmetry concepts as convexity, concavity, concentricity, containment and congruence. The idea of a "curvor" (a curved vector) is introduced and a precise mathematical definition of sharpness is given.

The relationship of sequential feature extraction to self- and mutually-referential languages is discussed and designs of electronic systems to perform efficient pattern analysis are outlined.

A CYBERNETICS OF SEEING – OR A FAST HIERARCHICAL ALGORITHM FOR COMPLEX BOUNDARY ENCODING TO ESTABLISH SIMPLE INFERENCE GRAMMARS OF A SELF AND MUTUALLY REFERENTIAL TYPE

1 INTRODUCTION

> "This process is perfectly adapted to mechanisation, and serves as a method to identify the shape of a figure independently of its size or its orientation or whatever transformation may be included in the group-region to be scanned."
>
> Norbert Wiener

For over 30 years now Cyberneticians have been concerned with the processes of seeing. Norbert Wiener, in his outstanding book – Cybernetics (1948) – devoted a complete chapter to seeing, entitled 'Gestalt and Universals' and thus laid the footings for us to build upon.

In this paper an attempt shall be made, in the context of my own current research, to expand and clarify Wiener's original ideas that he formulated around Locke's and Hume's philosophy of the association of ideas and the concept of group scanning – then excitingly new.

Wiener, together with other initiates such as Warren McCulloch, was perfectly aware that the process of seeing, with whatever sense it might be associated, was one that could be mechanised. That is to say, it is precisely and mathematically describable. To illustrate the point, Wiener wrote down a general integral:

$$\int Q(TS)dT \qquad (1)$$

where:

S is a set of elements transformed by the group

Q(S) is a quantity depending on S

and

T is a transformation of the group

which is a quantity that will remain fixed for all sets 'S' interchangeable with one another under the transformation of the group.

This formidable integral cannot be evaluated, however, unless the nature of operator 'Q' is known - strangely Wiener never expressly derived such an operator for any particular group. What follows is a partial derivation of 'Q' with its application to language, learning and thought.

2 THE INPUT PATTERN

I do not wish to belabour the fact already emphasised by Wiener and justified by Shannon that a continuous pattern can be mathematically represented at a point 'a' by F(a), where $a \in R \times R^*$ and $F(a) \in R$ and this in turn may be approximately transformed into a two-dimensional matrix: 'M' with elements

$$M(p) = F(kp + l) \tag{2}$$

where 'p' is an integer co-ordinate ($p \in I \times I^*$), 'k' is a real scaling factor ($k \in R$) and 'l' is a vector shift ($l \in R \times R$), for this is well known to anyone who has watched television and is thus implicitly, if not explicitly, aware of Shannon's sampling theorem. However, I do wish to point out that 'F(a)' and 'M(p)' are extraordinarily broad mathematical representations of most of our universe in terms of planar mappings or sections and as such may represent almost any physical situation (eg written language). In practice the order of 'M' is finite and hence the two components of 'p' are bounded. For example 'p' may be an integer spatial co-ordinate (x,y) and M(p) an average measure of light intensity, where the p's are evenly distributed over a rectangular region as is illustrated in Figure 1A for a capital letter 'A' for several levels of quantisation. The operation of quantisation is again one of approximation and is well understood; nevertheless in this paper it is worth pointing out that in the limiting case quantisation becomes binarisation (or single level thresholding), that is:

---

* R is the set of real numbers and I is the set of integers and 'x' is their cartesian product.

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	2	2	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	2	2	3	2	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	2	2	0	3	3	0	0	0	0	0	0	0
0	0	0	0	0	0	0	2	3	2	0	0	4	3	0	0	0	0	0	0
0	0	0	0	0	0	0	3	2	0	0	0	4	5	0	0	0	0	0	0
0	0	0	0	0	0	3	4	0	0	0	0	4	6	4	0	0	0	0	0
0	0	0	0	0	0	4	5	6	0	0	0	0	6	5	0	0	0	0	0
0	0	0	0	0	0	5	5	0	0	0	0	0	6	6	0	0	0	0	0
0	0	0	0	0	6	7	5	0	0	0	0	0	5	6	1	0	0	0	0
0	0	0	0	0	6	6	5	0	0	0	0	3	4	5	6	4	0	0	0
0	0	0	0	0	5	6	6	6	5	4	0	3	4	5	5	5	0	0	0
0	0	0	0	0	5	6	7	7	7	7	5	6	6	7	2	0	0	0	0
0	0	0	0	4	5	6	7	7	7	8	7	7	7	7	6	2	0	0	0
0	0	0	0	3	4	6	6	0	0	0	0	0	6	7	5	1	0	0	0
0	0	0	0	2	4	2	0	0	0	0	0	0	0	6	6	2	0	0	0
0	0	0	0	1	6	0	0	0	0	0	0	0	0	5	4	3	0	0	0
0	0	0	0	2	4	0	0	0	0	0	0	0	0	0	4	5	4	0	0
0	0	0	1	2	0	0	0	0	0	0	0	0	0	0	0	4	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

INPUT MATRIX 'M'

1A

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	1	1	0	1	1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	1	1	1	0	0	1	1	0	0	0	0	0	0
0	0	0	0	0	0	0	1	1	0	0	0	1	1	0	0	0	0	0	0
0	0	0	0	0	0	1	1	0	0	0	0	1	1	1	0	0	0	0	0
0	0	0	0	0	0	1	1	1	0	0	0	0	1	1	0	0	0	0	0
0	0	0	0	0	0	1	1	0	0	0	0	0	1	1	0	0	0	0	0
0	0	0	0	0	1	1	1	0	0	0	0	0	1	1	1	0	0	0	0
0	0	0	0	0	1	1	1	1	0	0	0	1	1	1	1	1	0	0	0
0	0	0	0	0	1	1	1	1	1	1	0	1	1	1	1	1	0	0	0
0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0
0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0
0	0	0	0	1	1	1	1	0	0	0	0	0	1	1	1	1	0	0	0
0	0	0	0	1	1	1	0	0	0	0	0	0	0	1	1	1	0	0	0
0	0	0	0	1	1	0	0	0	0	0	0	0	0	1	1	1	0	0	0
0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	1	1	0	0	0
0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

BINARISED MATRIX 'M_1'

1B

FIGURE 1

$$M_\theta(p) = B_\theta(M(p)) \tag{3}$$

where M_θ is a matrix with binary elements generated by applying an operator B_θ, which transforms all elements of M to '1' if they are above the value θ, ($\theta \epsilon R$), and to '0' otherwise – as illustrated in Figure 1B for the letter A. Clearly M may be represented by a finite set of M_θ's.

So far nothing new has been said – all that has been instanced is a particular method of deriving a member of Wiener's set 'S' and it is still necessary to describe mappings from 3D spaces to 2D planes with respect to visual perception, but I prefer to leave this pedestrian task until later and content myself with discussing contiguity before formally introducing my interpretation of 'Q'.

The adjective contiguous means touching, adjoining, near and it is in these words that the clue to the analysis and understanding of patterns lies. If a matrix 'M_θ' is given a random distribution of binary states it will, on average, possess a minimum of contiguity. That is, there will be no specifiable group or group transformations that will re-express M_θ, on average, in a more compact form – in other words, there are no properties either global or local in any sense "close" to a rectangular random distribution – by definition.

Fortunately for us the universe can be explained, for it abounds in repetition and thus 'Q' must detect conformity, continuity and order. In so doing, re-express M_θ in a more precise and understandable way.

3 THICKENING AND LOGICAL DIFFERENCING

Logical differencing is a simple mathematical operation 'L' on adjacent or neighbouring (contiguous) elements of 'M_θ' to emphasise and encode boundary information and test for regions of homogeneity (conformity) in M_θ – that is, clusters of '1's' or '0's'. Essentially adjacent binary elements of M_θ are operated on, as 2 × 2 sub-matrices of M_θ to produce numbers 0–8 defined by 'L' as in Table 1 and shown diagramatically in Figure 2A.

$\begin{matrix}0 & 0\\ 0 & 0\end{matrix} = 0 \qquad \begin{matrix}1 & 1\\ 1 & 1\end{matrix} = 0 \qquad \begin{matrix}1 & 0\\ 0 & 1\end{matrix} = 0 \qquad \begin{matrix}0 & 1\\ 1 & 0\end{matrix} = 0$

$\begin{matrix}1 & 0\\ 0 & 0\end{matrix} = 0 \qquad \begin{matrix}0 & 1\\ 1 & 1\end{matrix} = 2$

$\begin{matrix}0 & 1\\ 0 & 0\end{matrix} = 0 \qquad \begin{matrix}1 & 0\\ 1 & 1\end{matrix} = 8 \qquad \begin{matrix}1 & 1\\ 0 & 0\end{matrix} = 5 \qquad \begin{matrix}0 & 0\\ 1 & 1\end{matrix} = 1$

$\begin{matrix}0 & 0\\ 0 & 1\end{matrix} = 0 \qquad \begin{matrix}1 & 1\\ 1 & 0\end{matrix} = 6$

$\begin{matrix}0 & 0\\ 1 & 0\end{matrix} = 0 \qquad \begin{matrix}1 & 1\\ 0 & 1\end{matrix} = 4 \qquad \begin{matrix}1 & 0\\ 1 & 0\end{matrix} = 7 \qquad \begin{matrix}0 & 1\\ 0 & 1\end{matrix} = 3$

LOGICAL DIFFERENCE STATES

2A

0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	2	0	8	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	3	0	6	4	8	0	0	0	0	0	0	0
0	0	0	0	0	0	0	2	0	7	0	4	8	0	0	0	0	0	0
0	0	0	0	0	0	3	0	6	0	0	3	0	7	0	0	0	0	0
0	0	0	0	0	0	2	6	0	0	0	3	0	8	0	0	0	0	0
0	0	0	0	0	3	0	8	0	0	0	0	4	0	7	0	0	0	0
0	0	0	0	0	3	0	6	0	0	0	0	3	0	7	0	0	0	0
0	0	0	0	0	2	0	7	0	0	0	0	3	0	8	0	0	0	0
0	0	0	0	3	0	0	8	0	0	0	0	2	0	0	8	0	0	0
0	0	0	0	3	0	0	0	8	1	0	3	0	0	0	0	7	0	0
0	0	0	0	3	0	0	0	0	0	8	2	0	0	0	6	0	0	0
0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	8	0	0	0
0	0	0	3	0	0	0	6	5	5	5	5	4	0	0	0	7	0	0
0	0	0	3	0	0	6	0	0	0	0	0	0	4	0	0	7	0	0
0	0	0	3	0	6	0	0	0	0	0	0	0	3	0	0	7	0	0
0	0	0	3	0	7	0	0	0	0	0	0	0	0	4	0	8	0	0
0	0	0	2	6	0	0	0	0	0	0	0	0	0	0	4	6	0	0
0	0	0	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

LOGICALLY DIFFERENCED MATRIX

2B

FIGURE 2

$M_\theta(P_{x,y})$	$M_\theta(P_{x+1,y})$	$M_\theta(P_{x,y+1})$	$M_\theta(P_{x+1,y+1})$	$L(P_{x,y})$
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	3
0	1	1	0	0
0	1	1	1	2
1	0	0	0	0
1	0	0	1	0
1	0	1	0	7
1	0	1	1	8
1	1	0	0	5
1	1	0	1	4
1	1	1	0	6
1	1	1	1	0

TABLE 1

This operation 'L' on 'M_θ' is shown in Figure 2B for a letter 'A'. Notice that this is a 4 to 1 sub-mapping that reduces the total dimension of the input matrix by one, yet increases the number of elemental states to 9. The transition is primarily intended to extract boundary information in an immediately useful form and is easily implemented by either a "look-up" table (eg a ROM) or combinatorial logic. The idea is not totally new and stems from the arithmetical differencing nets proposed by Golay and others.

Little information is lost in making the transformation:

$$L(M_\theta) = D_\theta \qquad (4)$$

only lone single points and lone diagonal pairs of points being set to zero and thus all other situations being recoverable by the inverse operation:

$$M_\theta' = L^{-1}(D_\theta) \qquad (5)$$

Solid regions are refillable with '1's' by a test to be described in ~~the section on~~ vector chains.

In order to facilitate the analysis of fine diagonal lines (lines less than the dimensions spanned by a 2 x 2 sub-matrix) as are shown in Figure 3A, a thickening operation, as is shown in Figure 3B. can be performed before logical differencing takes place, according to the following re-write rules:

$$M''_\theta(P_{xy}) = M''_\theta(P_{x+1,y+1}) = \left\lceil M_\theta(P_{x,y}) = M_\theta(P_{x+1,y+1}) = 0, M(P_{x+1,y}) = M(P_{x,y+1}) \right\rceil = 1 \quad + M(P_{xy}$$

$$M''_\theta(P_{x,y}) = M''_\theta(P_{x-1,y+1}) = \left\lceil M_\theta(P_{x-1,y}) = M_\theta(P_{x,y+1}) = 1, M(P_{x,y}) = M(P_{x-1,y+1}) \right\rceil = 0 \quad + M(P_{x,}$$

(6)

where '$\lceil e \rceil$' takes the value 1 (zero otherwise) if 'e' is true. Thus an irreversible operator 't' may first be applied to M_θ, that is

$$M''_\theta = t(M_\theta) \tag{7}$$

followed by the logical differencing operation 'L' to produce:

$$D_\theta = L(t(M_\theta)) \tag{8}$$

which is only slightly "lossy" in information since M_θ can be almost totally reconstructed from D_θ.

4 BOUNDARY CHAIN ENCODING

The general problem of following a contour of a two-dimensional binarised shape has been well researched.

However there have remained certain difficulties in that:

a Inner contours are difficult to find, especially if the 'a priori' information about the shapes to be followed is low.

b Markers normally have to be left about the followed pattern, so that a border is not cycled round indefinitely.

and

c Particular care has to be taken to ensure a previously followed shape is not returned to - meaning special tests have to be applied to prevent this.

```
                      0
                    0 1 0
                  0 1 0 1 0
                0 1 0   0 1 0
              0 1 0       0 1 0
            0 1 0           0 1 0
          0 1 0               0 1 0
        0 1 0                   0 1 0
      0 1 0                       0
    0 1 0
  0 1 0
0 1 0
  0
```

$$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

A THICKENING SITUATION

3A

```
                      0
                    1 1 1
                  1 1 1 1 1
                1 1 1 0 1 1 1
              1 1 1       1 1 1
            1 1 1           1 1 1
          1 1 1               1 1 1
        1 1 1                   1 1 0
      1 1 1                       0
    1 1 1
  1 1 1
0 1 1
  0
```

$$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \Rightarrow \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \Rightarrow \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$$

A THICKENING OPERATION

3B

<u>FIGURE 3</u>

The technique to be described helps solve these difficulties by extracting relative information that can be made independent of

1 Translation
2 Size
3 Rotation
and
4 Perspective

that is, exactly those qualities demanded by Wiener's transformation operator 'T' on the set 'S'. This will now be shown to be accomplished by a combination of the logical differencing procedure, described in the previous sections, that may incidentally be executed in parallel, and a sequential contour following process which collectively produces sets of vector chains that can be greatly simplified by a vector association routine. To finally arrive at a collection of curved polygonal approximations to the pattern that may be further classified by linguistic techniques.

4a STATE ASSOCIATION

As will be seen from Figure 4A, it is possible to associate 8 two-dimensional vectors with the states '1' to '8'. These vectors are in the directions of the 8 points of the compass (E, NE ... SE) and are either 1 or $\sqrt{2}$ units long according to their bearing (ie E, N, W and S are 1 unit long, otherwise $\sqrt{2}$ units) - this common type of boundary encoding was originally used by Freeman (1961).

It will be realised that only certain vectors can join up with certain other vectors in other neighbouring cells (Figure 4B) and it is this fact alone that allows for their sequential collection by a special operator 'C' which performs the algorithm summarised by the Flow Chart 1.

Several points are worth emphasising about this algorithm before discussing the collection process mathematically. They are:

a The algorithm performs a raster scanning - left to right, top to bottom - until a non-zero state is found,

when

b The contour is then traced round, recording the co-ordinates and state of the first non-zero cell. then zeroing this and subsequent cells (after their state has been recorded) as they are left, thus ensuring the contour is not returned to later.

This continues until

c The first zero cell is reached when the raster scanning process is returned to until either an inner contour is found or another outer boundary is met or the bottom right-hand cell is arrived at, when the process stops.

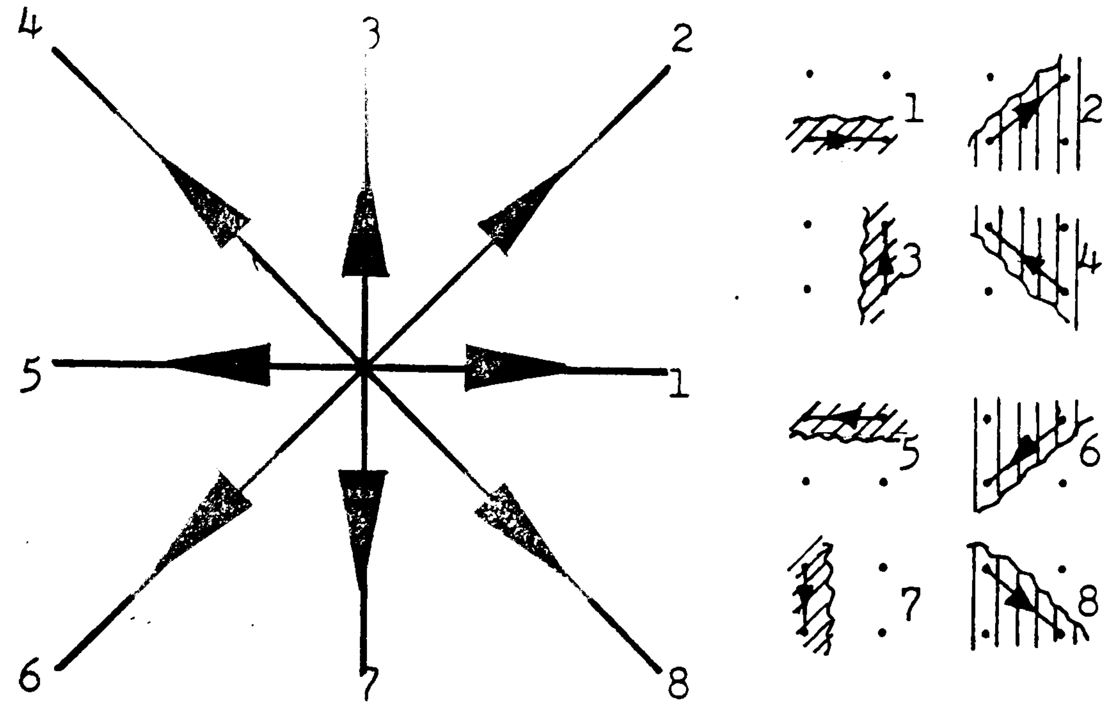
Thus what is achieved is a reformulation of the binary matrix M_B in terms of a set of vector chains 'C', each vector chain 'V_R' beginning at the top left-hand corner of the shape, this co-ordinate initially being recorded and all other points being relative to this point.

The actual association of states is achieved by a combination of a next possible address increment function 'dp' and a state compatibility function 'c' as are given below in Tables 2 and 3.

	Present State 's'	Search Direction No.	Increment Function 'dp'
x-direction	1	1 (ie horizontal)	1 (ie move right)*
	2	1	1
	3	1	1
	4	1	-1 (ie move left)*
	5	1	-1
	6	1	-1
	7	1	-1
	8	1	1
y-direction	1	2 (ie vertical)	1 (ie move down)*
	2	2	-1 (ie move up)*
	3	2	-1
	4	2	-1
	5	2	-1
	6	2	1
	7	2	1
	8	2	1

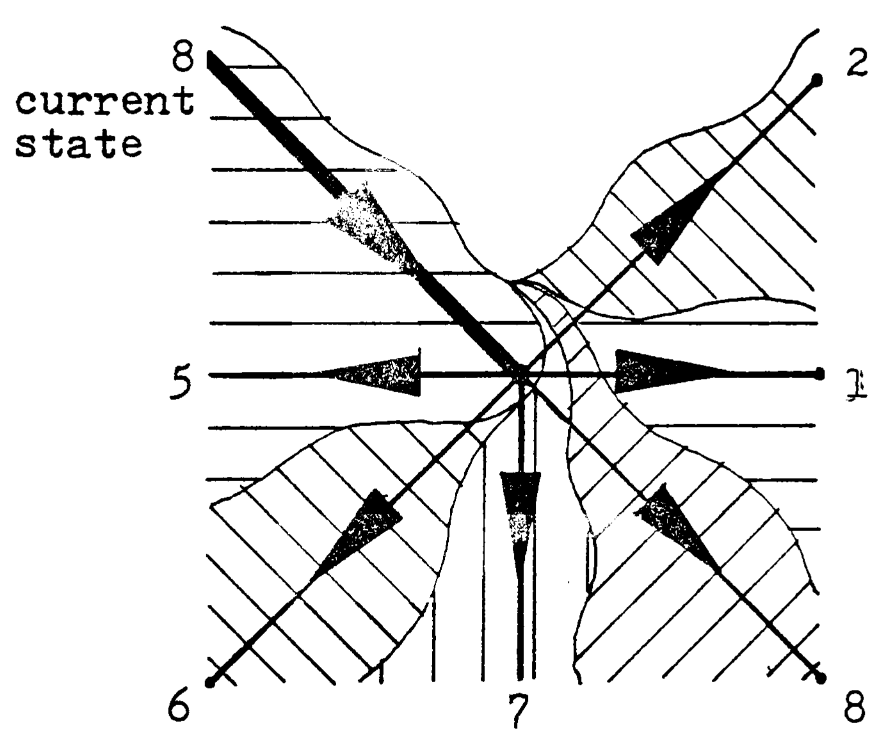
TABLE 2

*Comments refer to normal matrix addressing



VECTOR STATES

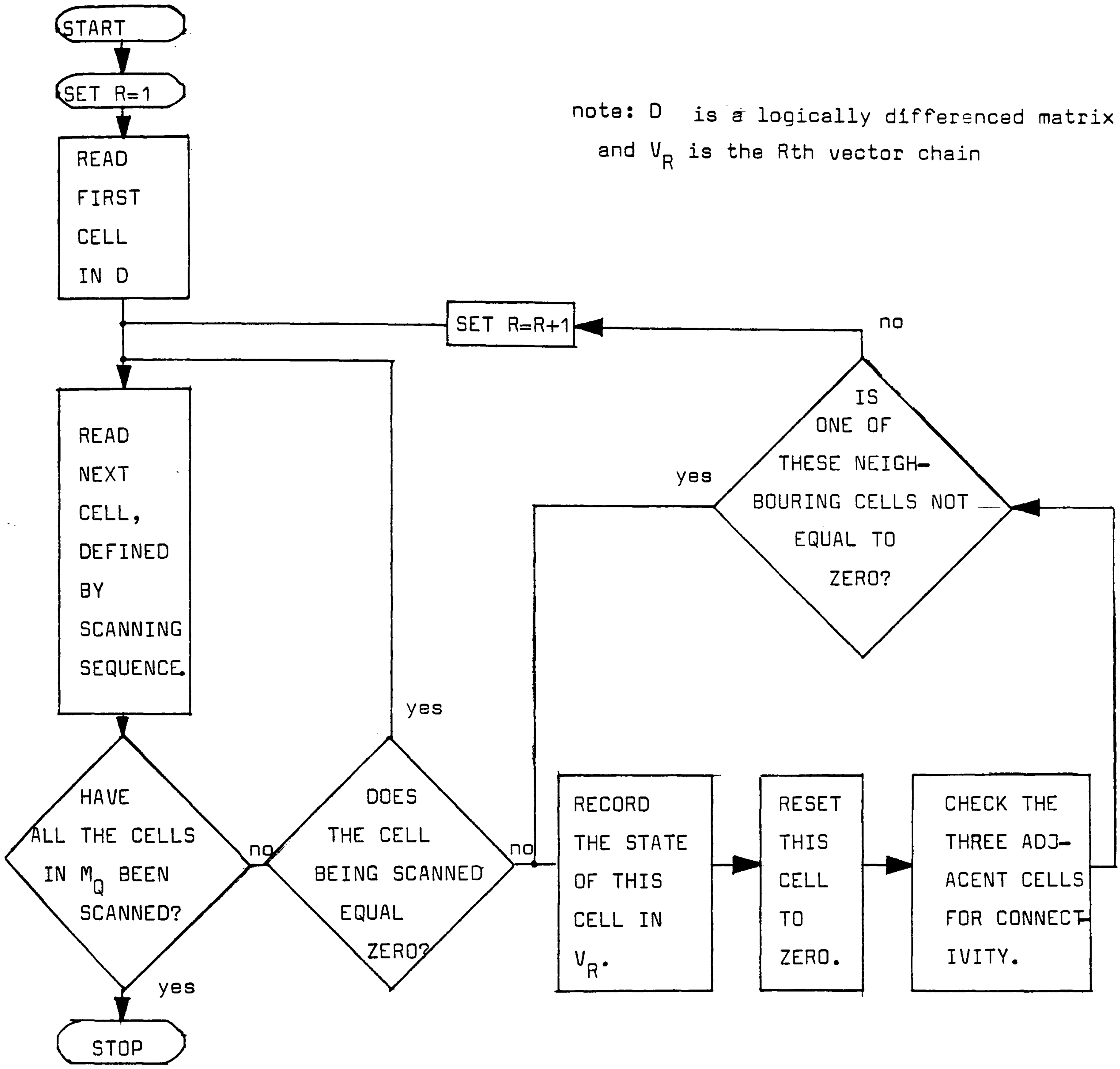
4A



VECTOR STATE COMPATIBILITY

4B

FIGURE 4



note: D is a logically differenced matrix and V_R is the Rth vector chain

A FLOW CHART FOR CONTOUR FOLLOWING

FLOW CHART 1

Present State 's'	Alternative State No.	Alternative Cell No.	State Compatibility Function 'c'
1	q	1 (ie vertical)	1+q
2	q	1	7+q
3	q	1	7+q
4	q	1	5+q
5	q	1	5+q
6	q	1	3+q
7	q	1	3+q
8	q	1	1+q
1	q	2 (ie diagonal)	5+q
2	q	2	3+q
3	q	2	3+q
4	q	2	1+q
5	q	2	1+q
6	q	2	7+q
7	q	2	7+q
8	q	2	5+q
1	q	3 (ie horizontal)	7+q
2	q	3	1+q
3	q	3	1+q
4	q	3	3+q
5	q	3	3+q
6	q	3	5+q
7	q	3	5+q
8	q	3	7+q

where q = '0' or '1'

TABLE 3

Notice that 'dp' dictates where to try next and is primarily a function of the previous state and secondly the direction of search, which must be one of three choices (1, 2 and '1 and 2') - since a state vector can only link with three other cells, the third choice being an inclusion of choices 1 and 2 of the search direction number to give a diagonal step. Strictly dp is a two-dimensional vector, but to illustrate efficient storage the x and y components have been separated in Table 2.

Similarly 'c' is a function of the state 's' and also includes the option of one of two possible neighbours per cell in one of three alternative cells dictated by 'dp'. Thus when 'dp' points to the next cell 'c' is applied to the previous cell and if the two are compatible the new state is included in the vector chain V_R and the boundary tracing continues, otherwise another of the three options is tried until a compatible state is found or the raster scan is allowed to continue and the chain closed.

When all the cells have been scanned then

$$C(D_\theta) = \left\{V_1, V_2, \ldots V_R, \ldots V_N\right\} \tag{9}$$

where 'C' is a state association operator and 'V_R' is the Rth vector chain, that is:

$$V_R = \left\{s_{1,R}, s_{2,R}, \ldots, s_{n(R),R}\right\} \tag{10}$$

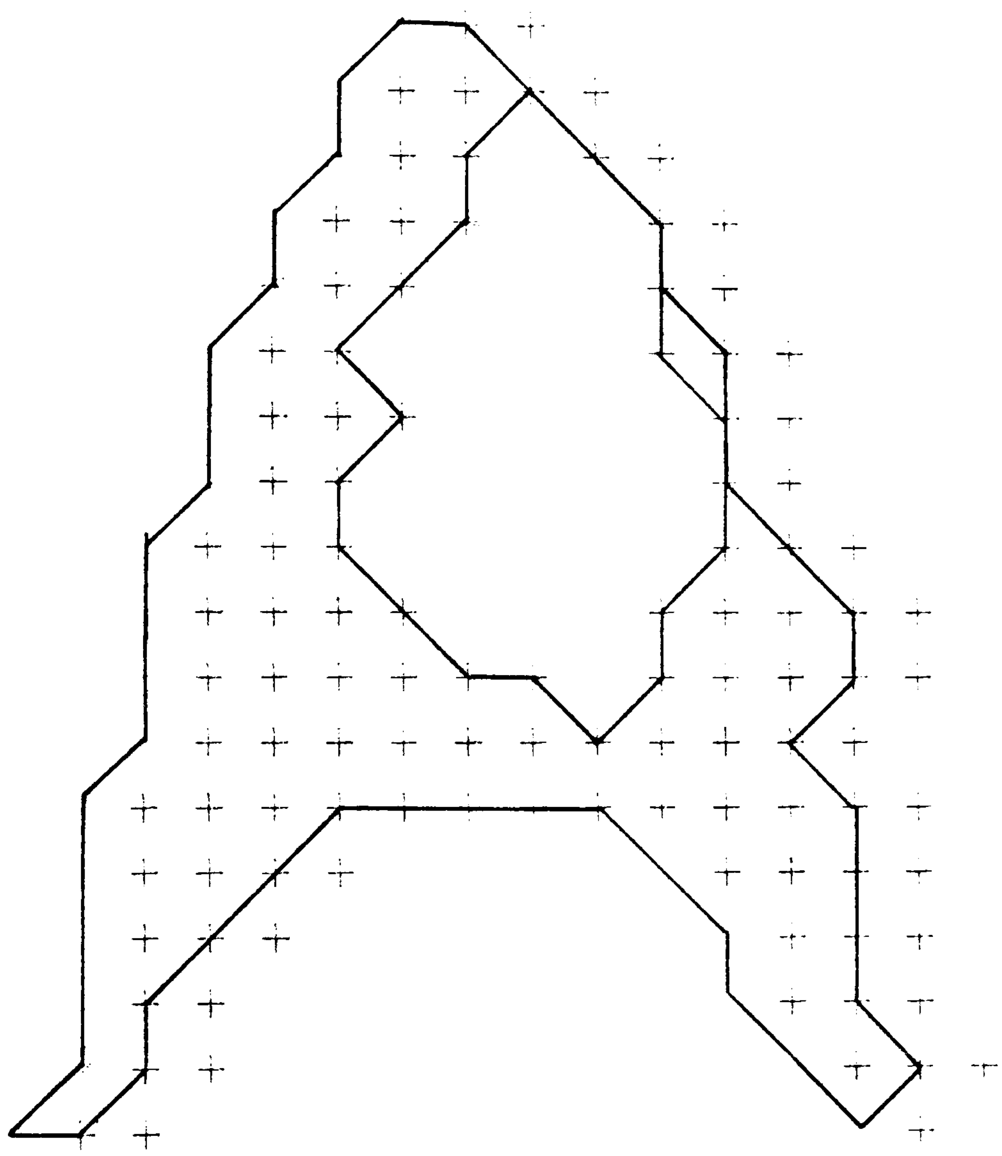
where s_{rR} is the rth vector state collected in the V_Rth chain (ie an integer in the range 0–8) and n(R) is the number of vector states in the Rth vector chain. An illustration of $C(D_\theta)$ is given in Figure 5A for the letter 'A' of Figure 1.

Edge effects are easily dealt with by having an extended border of permanently zeroed cells around M_θ, before 'L' is applied. In this way closed shapes are always realised within the total pattern – the straight boundary edges can later be replaced by the actual contour outside the sampling matrix 'M' when 'M' is moved or appropriately altered (eg zoomed out) to include the rest of the contour. This "stitching together" of contours is an important compromise in the visual world that allows a scene to be analysed through a full 360 degrees.

4b VECTOR ASSOCIATION

In order to reduce the complexity of vector chains, various associative operations may be devised.

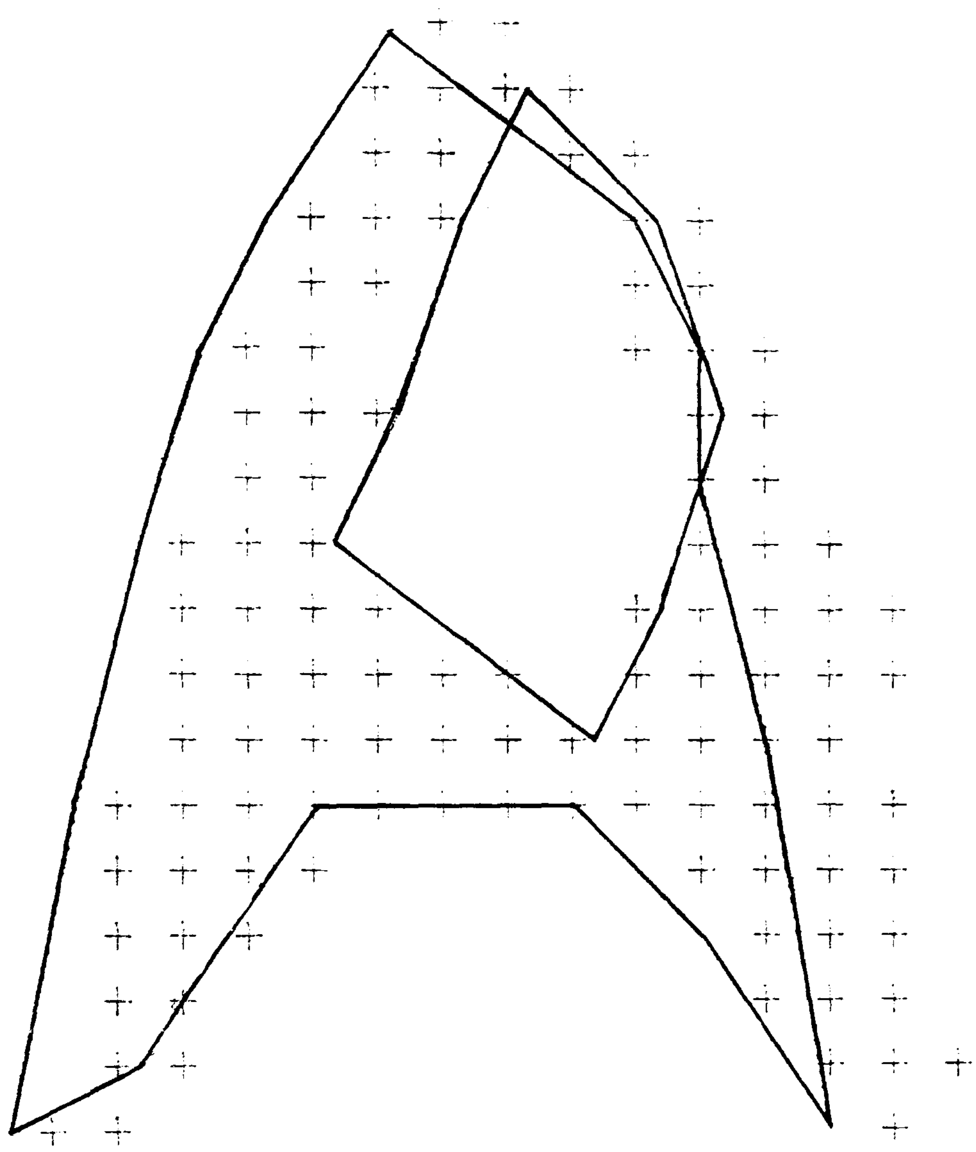
If state '1' is represented as a two-dimensional vector (1,0), state '2' as (1,1) and so on, anticlockwise round the compass, to state '8' (1,-1), (see Figure 4A) then it is possible to associate the vector elements of V_R together by addition, that is



VECTOR STATE CHAIN

5A

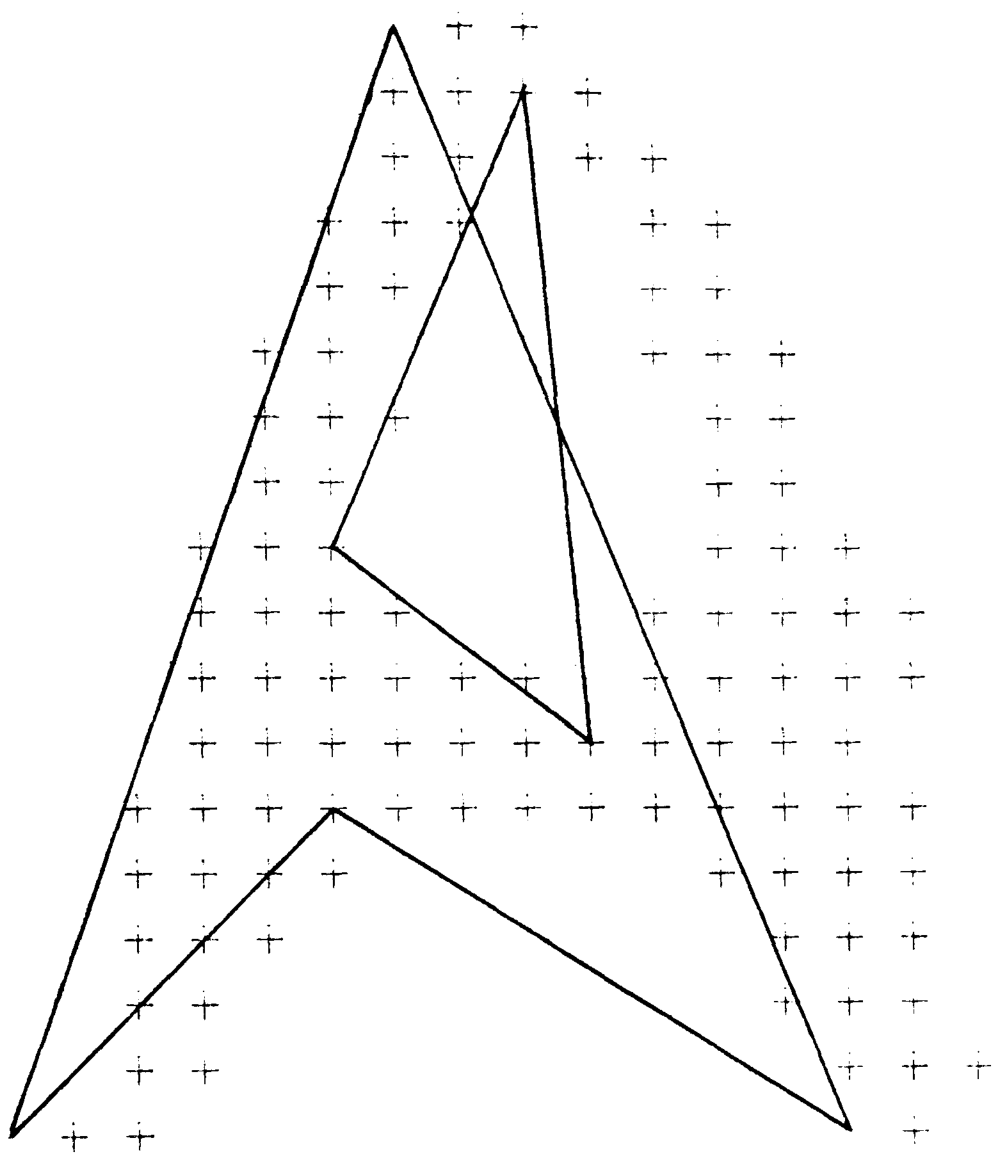
FIGURE 5



VECTOR CHAIN OF LOW ASSOCIATION

5B

FIGURE 5



VECTOR CHAIN OF HIGH ASSOCIATION

5C

FIGURE 5

$$V_R^{'} = A(V_R) = \left\{ \sum_{r=1}^{r=n1} V_{Rr}, \sum_{r=n1}^{r=n2} V_{Rr}, \ldots \sum_{r=n(R)-m}^{r=n(R)} V_{Rr} \Big/ 1 \leq n1 \leq n2 \ldots \leq n(R)-m \leq n(R) \right\} \quad (11)$$

where 'A' is a vector associative operator and 'v_{Rr}' is a vector representation of state 's_{Rr}'. The rules by which these local summations are to be carried out shall now be discussed.

The simplest criterion by which two vectors may be associated together is their similarity (equivalence), thus if two vectors point in the same direction they can be added without significant loss of information and this procedure will extract horizontal, vertical and 45^{o} diagonal straight lines. Let this associative operation be represented by A_e acting on V_R

$$V_R^{''} = A_e(V_R) \quad (12)$$

To extract straight lines of other inclination the incremental transitions have to be searched for and this is achieved by adding single compatible vectors to the straight line regions already established by the equivalence criterion applied by A_e. The compatible vectors are given by the relation:

$$s_c = s+q \text{ (modulo 8) for } q = 7 \text{ or } 1 \quad (13)$$

where 's_c' is a compatible state and 's' is the current state.

Let this associative operation be represented by A_i acting on $V_R^{''}$

$$V_R^{'''} = A_i(V_R^{''}) \quad (14)$$

Thirdly an association of vectors in $V_R^{'''}$ can be based on a tolerance angle 'δ' established by taking the inner '$\bullet$' and outer products '$\wedge$' to evaluate the signed angles between consecutive elements of $V_R^{'''}$, that is:

$$\angle(v_{R,r}^{'''}, v_{R,r+1}^{'''}) = \operatorname{sgn}(v_{R,r}^{'''} \wedge v_{R,r+1}^{'''}) \arccos\left(\frac{v_{Rr}^{'''} \bullet v_{R,r+1}^{'''}}{|v_{R,r}^{'''}||v_{R,r+1}^{'''}|} \right) \quad (15)$$

where '$\angle$' is an angle operator defined over $+180^0$ and -180^0 'sgn' takes the sign of the number it operates on ($\mathrm{sgn}(e) = 1$ if $e \geq 0$, otherwise -1) and $|v|$ take the Euclidean length of 'v'. Thus if criterion

$$-\delta \leq \angle(v'''_{Rr}, v'''_{Rr+1}) \leq \delta \qquad (16)$$

is satisfied then $v'''_{R,r}$ and $v'''_{R,r+1}$ may be added together. This may be represented by a third associative operator A_δ, that is:

$$V_R^{IV} = A_\delta\,(V_R''') \qquad (17)$$

It should be noted that A_δ is a special case of A_e when $\delta = 0$.

In summary, A_e associates equivalent vectors, A_i extends A_e to include increments and A_δ straight lines within a tolerance 'δ'. A fourth possibility 'A_λ' is to add in the very short vectors (ie $|v_{Rr}| \leq \lambda$) as these may be just noise. Thus

$$V_R' = A(V_R) = A\ (A\ (A_i(A_e(V_R))) = \left\{ v_{R,1}', v_{R,2}' \cdots v_{R,n}' \right\} \qquad (18)$$

is the total vector association and produces the representations shown in Figures 5B and 5C for the same letter A. Clearly the parameter 'δ' and 'λ' determine how much information is "thrown away" and how many vectors are left.

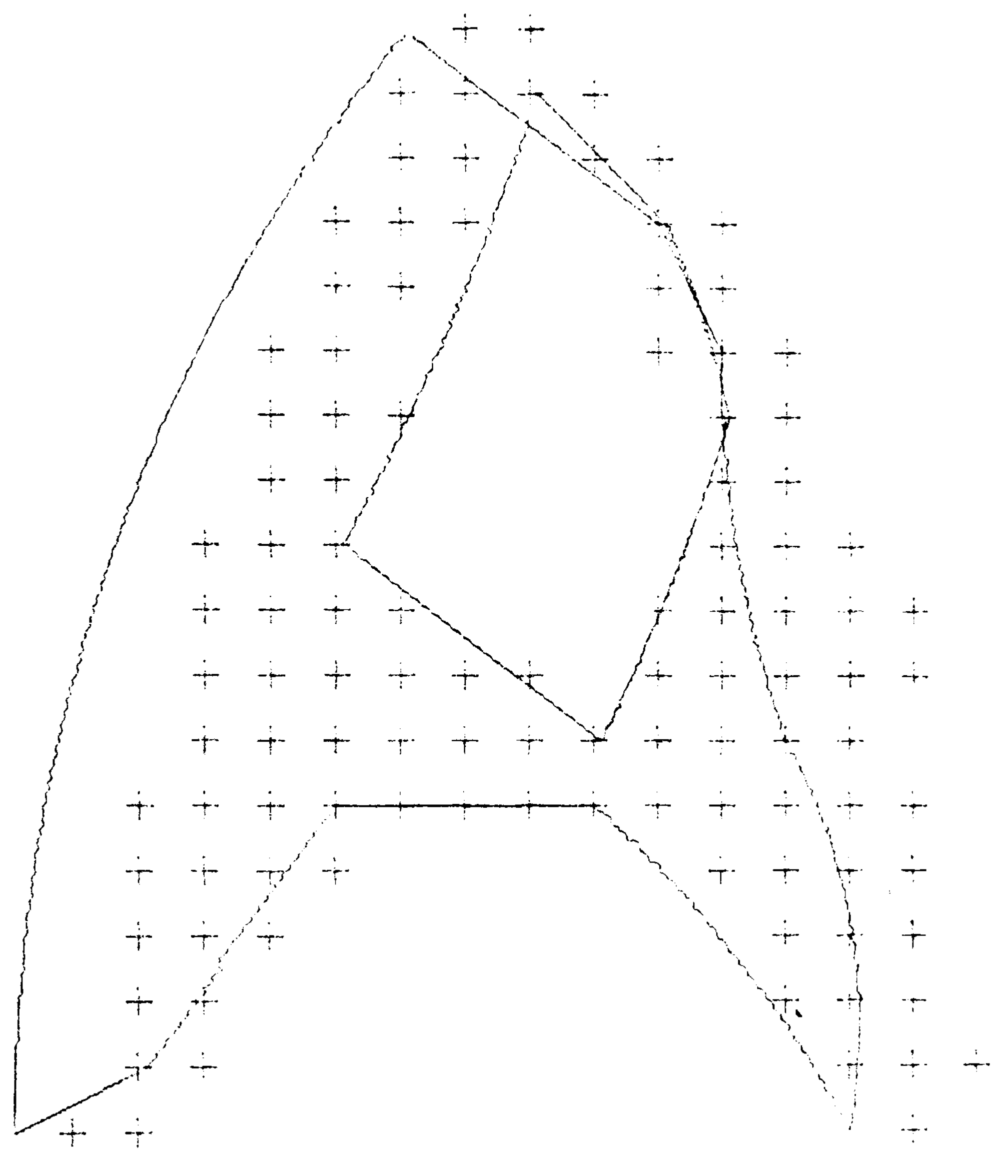
4c CURVOR ASSOCIATION

In order to further reduce the complexity of a vector chain 'V_R' it is necessary not only 'to recognise' straightness, but also curvature, and this can be achieved by re-expressing V_R' as "curvor chain" -"X"

A curvor is defined to be a curved vector that is:

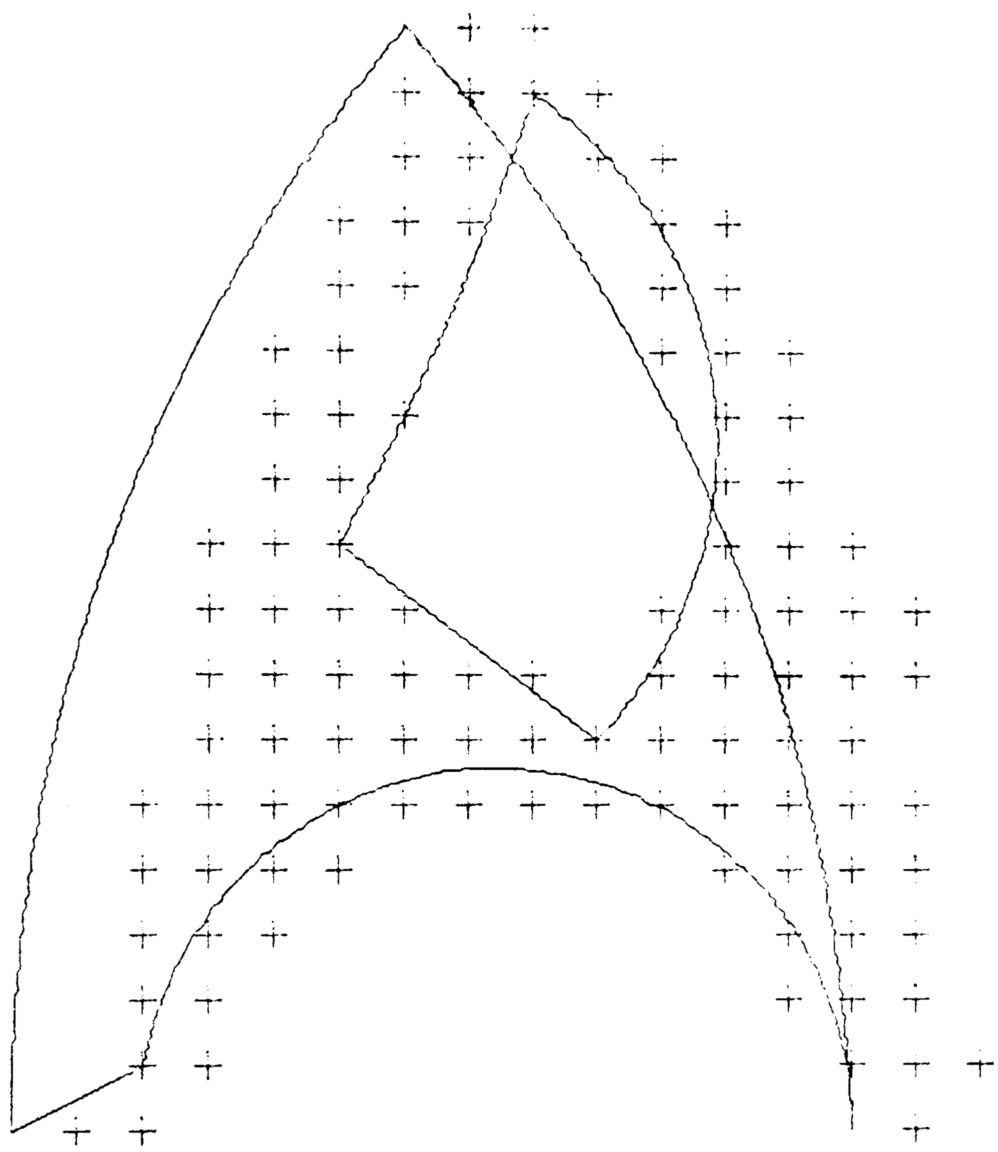
$$X = \begin{pmatrix} v \\ \varkappa \end{pmatrix} = \begin{pmatrix} x \\ y \\ \varkappa \end{pmatrix} \qquad (19)$$

where x and y are elements of 'v' and '$\varkappa$' is a fixed angle subtended by two lines from the two ends of v to a free point that thus describes - as a locus - a unique curved line of constant radius of curvature as is shown



CURVOR CHAIN WITH SMALL TOLERANCES

FIGURE 7A



CURVOR CHAIN WITH HIGH TOLERANCES

FIGURE 7B

in Figure 6A. '$\varkappa$' shall be known as the angle of curvature. In practice two vectors v_a and v_b may be 'added' together (represented by $\not+$) to form an initial curvor:

$$X = v_a \not+ v_b = \begin{pmatrix} v_a + v_b \\ \angle v_a, v_b \end{pmatrix} = \begin{pmatrix} v \\ \varkappa \end{pmatrix} \tag{20}$$

where $\angle v_a v_b$ is defined by expression 15. Clearly a tolerance τ must be imposed on this addition and is a parameter similar to 'δ' and 'λ' of the previous section. Any initial curvor may be <u>extended</u> by a vector v_c as is shown in Figure 6B, that is:

$$X' = X \not+ v_c = \begin{pmatrix} v + v_c \\ \angle v, v_c \end{pmatrix} \tag{21}$$

provided the following condition is satisfied:

$$| \angle v_a, v_b + v_c - \angle v_a + v_b, v_c | < \rho \tag{21a}$$

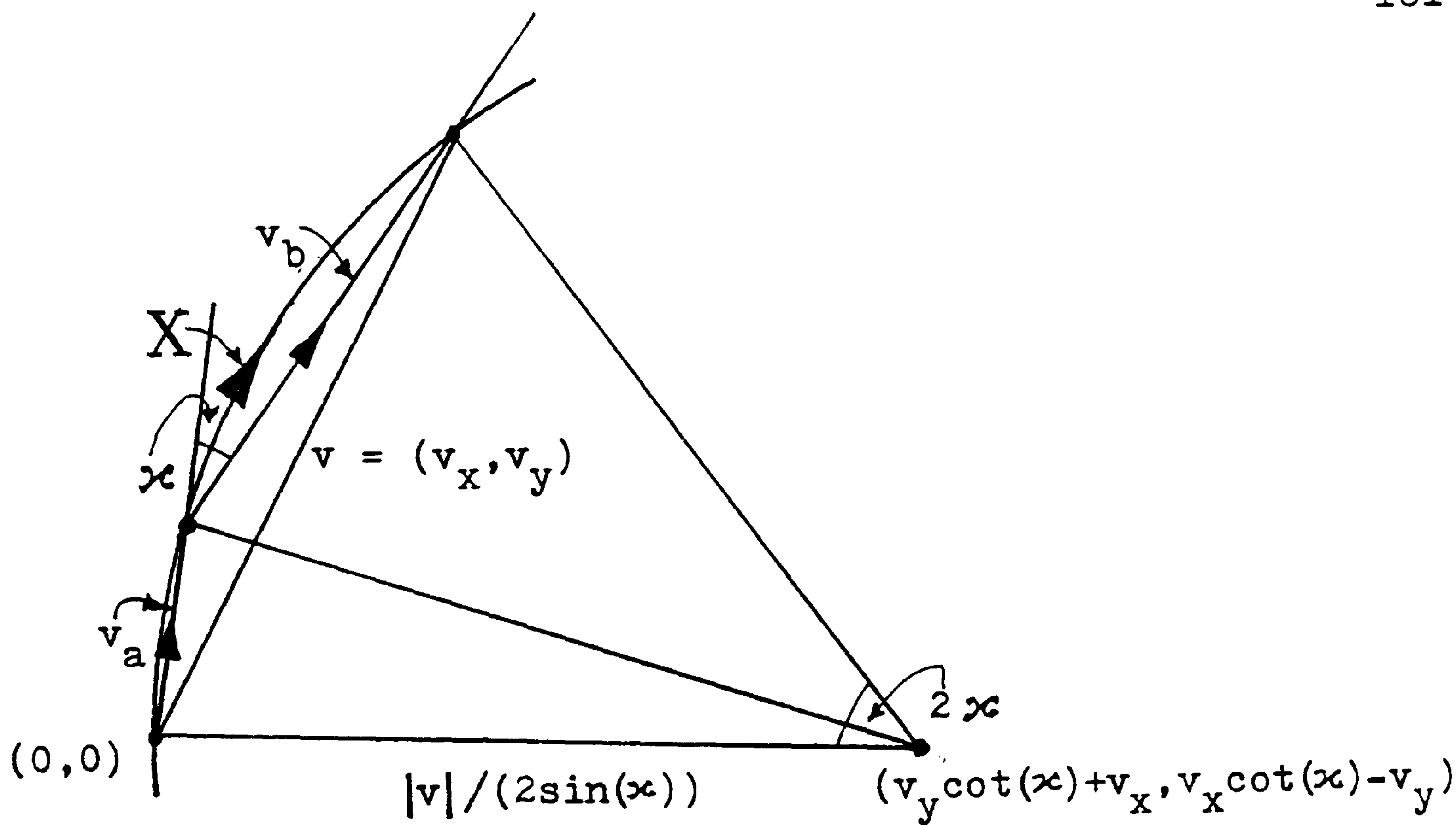
where expression 21a is a sufficient and necessary condition for constant curvature within a tolerance $\pm \rho$.

Clearly, as with straight line association, expression 21 and its condition may be applied recursively so that a vector chain V_R' may be operated on by 'χ' to produce

$$\chi(V') = \{X_1, X_2 \ldots X_n\} \tag{22}$$

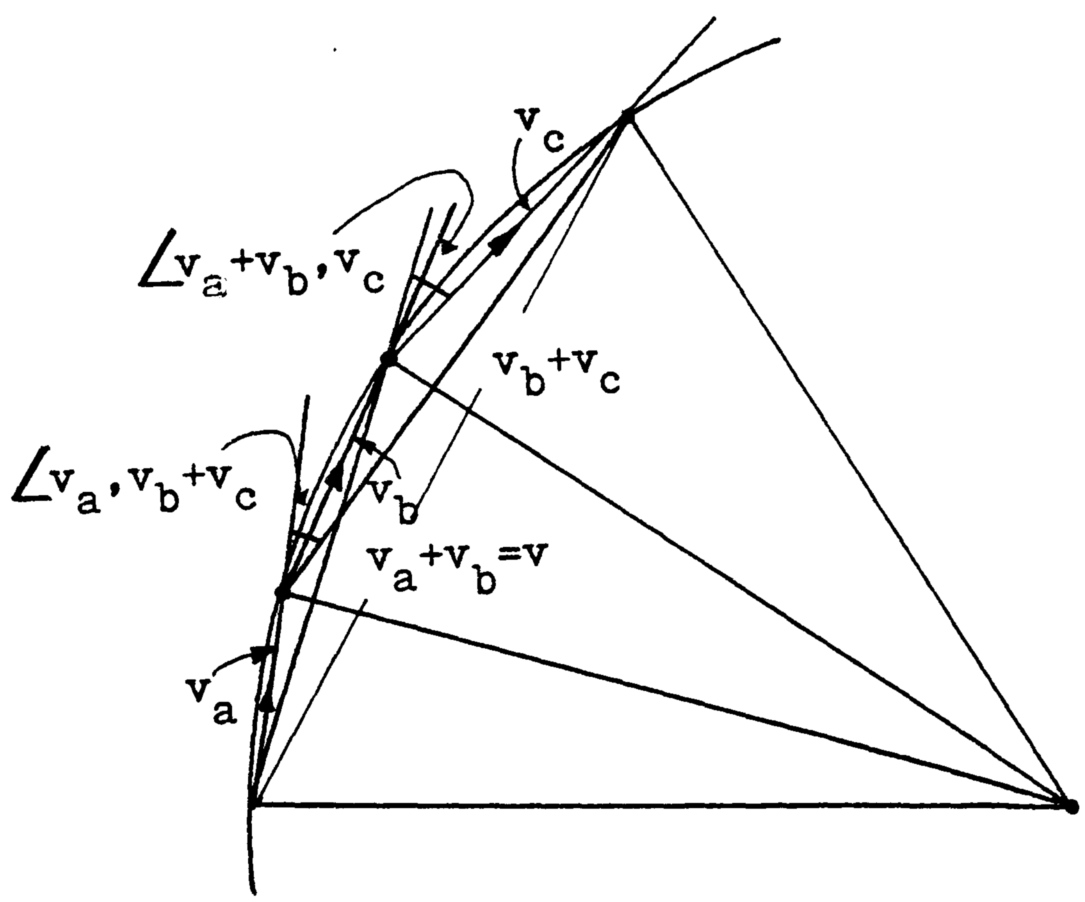
where X_r may be a straight line ($\varkappa = 0$), a convex curve ($\varkappa$ + ve) or a concave curve ($\varkappa$ -ve).

Examples of the operators 'χ' and 'A' applied $C(D_8)$ for the letter 'A' used in previous instances, are shown in Figures 7A and 7B, and illustrates the effects of parameters 'τ' and 'ρ'



CURVOR REPRESENTATIONS

6A



CURVOR EXTENSIONS

6B

<u>FIGURE 6</u>

Finally notice for completeness if:

$$X_r \not{+} X_{r+1} = \begin{pmatrix} v_r + v_{r+1} \\ \angle v_r, v_{r+1} \end{pmatrix} \tag{23}$$

and

$$X_r + X_{r+1} = \begin{pmatrix} v_r + v_{r+1} \\ \varkappa_r + \varkappa_{r+1} \end{pmatrix} \tag{24}$$

then '$\not{+}$' and '+' are identical if

$$\angle v_r, v_{r+1} = \varkappa_r + \varkappa_{r+1} \tag{25}$$

Thus curvors may be <u>added</u> without loss of information if expression (25) is satisfied or with some loss of information if

$$(\varkappa_r + \varkappa_{r+1}) - \rho \leq \angle v_r, v_{r+1} \leq (\varkappa_r + \varkappa_{r+1}) + \rho \tag{26}$$

5 SOME GEOMETRIC PROPERTIES OF VECTOR AND CURVOR CHAINS

Before continuing with the further reductive coding of boundary information in increasing useful forms, it is necessary to briefly look at some of the geometric properties of vector and curvor chains, in terms of their dimensionalities, their transformations and their universalities. This will allow relative and absolute languages to be established to describe and translate a wide variety of input patterns and give insight into Wiener's operator 'T'.

However, it should first be mentioned that a chain 'Φ' has certain inherent properties in that it is cyclic, that is:

$$\Phi = \{\varphi_1, \varphi_2 \ldots \varphi_n\}$$

$$\Phi = \{\varphi_n, \varphi_1 \ldots \varphi_{n-1}\}$$

$$\Phi = \{\varphi_{n-1}, {}_n \ldots \varphi_{n-2}\}$$

and so on until

$$\Phi = \{\varphi_2 \qquad \varphi_1\} \tag{27}$$

and in this way is a more precise concept than a string. Notice that vector chains are slightly redundant, in that their last element may always be deduced, since

$$\sum_{r=1}^{r=n} v_r = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \qquad (28)$$

which is clearly not true for curvor chains, since x_n cannot be deduced.

5a DIMENSIONALITIES

Certain properties such as width, length, area and perimeter are often used to describe and qualify physical entities; these shall now be discussed in abstract relation to vector and curvor chains.

5a i THE BOX PARAMETERS OF A VECTOR CHAIN

It is often useful to re-express a vector chain 'V' as:

$$\begin{aligned} V_p &= \left\{ v_1,\ v_1+v_2 \cdots \sum_{s=1}^{s=r} v_s \cdots \sum_{r=1}^{r=n} v_r \right\} \\ &= \left\{ v_{p1}, v_{p2} \cdots v_{pr} \cdots v_{pn} \right\} \end{aligned} \qquad (29)$$

where 'V_p' is an ordered set of position vectors as are shown in Figure 8.

Two useful operators on V_p are:

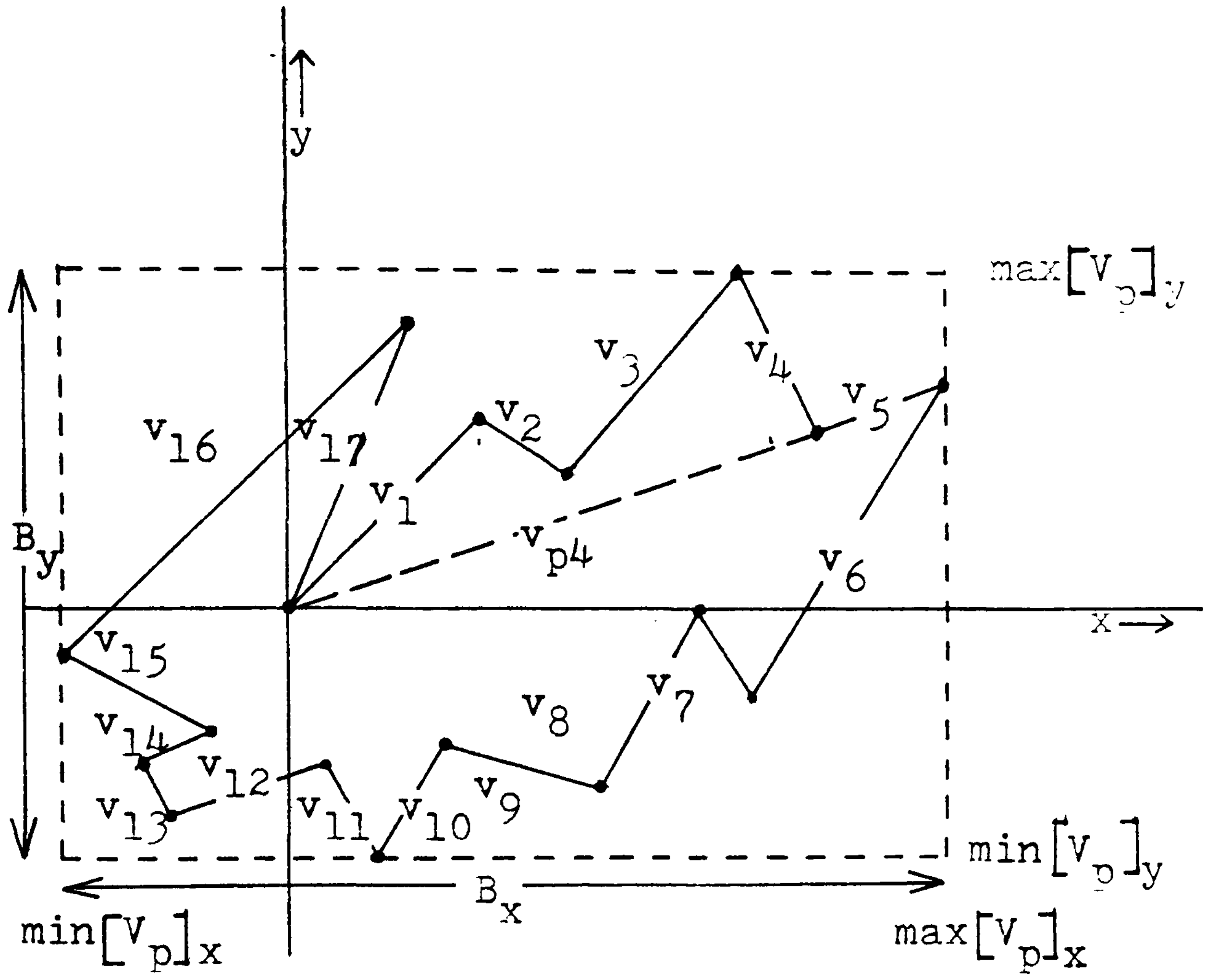
$$\max \left[V_p \right] = \begin{pmatrix} \text{maximum 'x' co-ordinate of } V_p \\ \text{maximum 'y' co-ordinate of } V_p \end{pmatrix} \qquad (30)$$

and

$$\min \left[V_p \right] = \begin{pmatrix} \text{minimum 'x' co-ordinate of } V_p \\ \text{minimum 'y' co-ordinate of } V_p \end{pmatrix} \qquad (31)$$

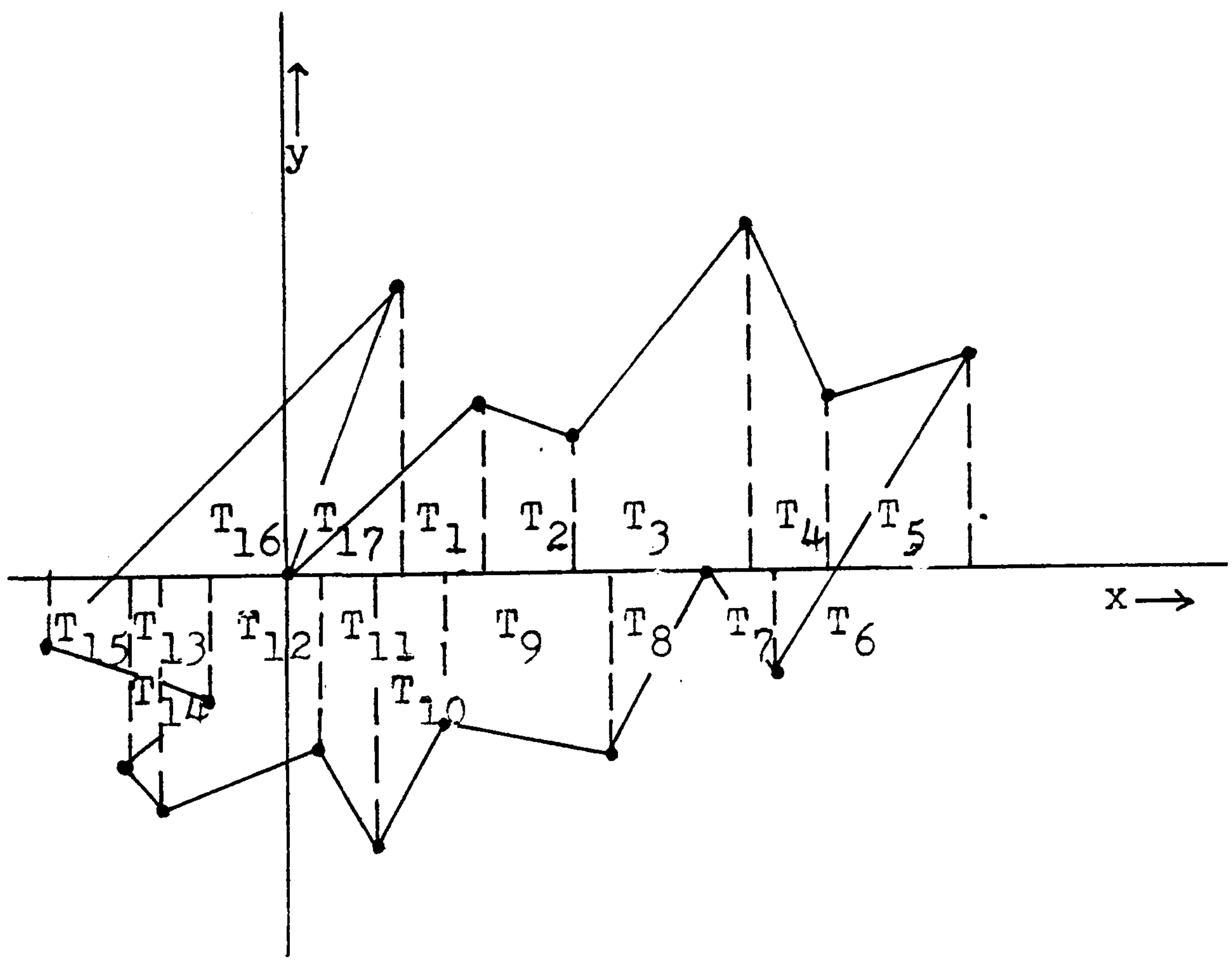
from which the box parameters 'B' of a vector chain 'V' may be defined as:

$$B(V) = \max \left[V_p \right] - \min \left[V_p \right] = \begin{pmatrix} B_x \\ B_y \end{pmatrix} \qquad (32)$$



A CHAIN'S POSITION VECTORS AND BOX PARAMETERS

8A



TRAPEZOIDAL SECTIONS

8B

<u>FIGURE 8</u>

which correspond to the 'horizontal' (ie row) and 'vertical' (ie column) extents of the vector chain relative to the sampling matrix 'M' as is also shown in Figure 8.

5a ii THE AREA ENCLOSED BY A VECTOR CHAIN

By adding together signed trapezoidal areas between an arbitrary reference line (eg the x or y axis) and the vector chains components (see Figure 8), it is possible to establish the area A_c contained within a vector chain as

$$A_c(V) = \tfrac{1}{2}\sum_{r=1}^{r=n} (v_r)_x((v_{p,r-1})_y + (v_{p,r})_y) \qquad (33)$$

$$= \tfrac{1}{2}\sum_{r=1}^{r=n} (v_r)_y((v_{p,r-1})_x + (v_{p,r})_x)$$

where $(v)_x$ and $(v)_y$ means take the 'x' and 'y' components of vector 'v' respectively.

5a iii THE PERIMETER OF A VECTOR CHAIN

Simply by summing the lengths of the elements of 'V' can its perimeter 'P' be evaluated, that is

$$P(V) = \sum_{r=1}^{r=n} |v_r| \qquad (34)$$

where '$|\;|$' has already been defined in section 4b.

5a iv SOME CORRECTIONS FOR CURVOR CHAINS

The box parameters are difficult to establish for a curvor chain 'χ' and should therefore have been previously evaluated for the vector chain from which it was derived. However, the perimeter 'P' of a curvor chain is easily found by:

$$P(\chi) = \sum_{r=1}^{r=n} f_1(\chi_r)\ |v_r| \qquad (35)$$

where

$$f_1(\chi) = \chi/_{\sin\chi} \qquad (35a)$$

Similarly the area contained by a curvor chain may be readily found by an extension of the linear case:

$$A_c(X) = A_c(V) + \sum_{r=1}^{r=n} f_2(\chi) \left|v_r\right|^2 \qquad (36)$$

where

$$f_2(\chi) = \frac{\chi - \sin\chi}{4\sin^2\chi} \qquad (36a)$$

In both cases, $f_1(\chi)$ and $f_2(\chi)$ may be a stored set of values, thus eliminating the necessity to test for $\chi = 0$. However, for the limiting cases $\chi = \pi^c$ both functions tend to infinity - fortunately appropriate tests in the curvor association process can prevent this occurring.

5b TRANSFORMATIONS

In this section special transformations of vector and curvor chains will be discussed, such as: contractions and dilations, translations, rotations and projections. These will be seen to fall into two classes of operator - linear and non-linear. Please note the invariances of the angle of curvature 'χ'.

5b i CONTRACTIONS AND DILATIONS

The box dimension of a vector chain may be changed linearly and uniformly by simply multiplying 'V' by a scalar 'μ':

$$V_\mu = \mu(V) = \mu\left\{v_1, v_2 \dots v_n\right\} \qquad (37)$$

$$= \left\{\mu v_1, \mu v_2 \dots \mu v_n\right\}$$

if $\mu \leq 1$ then the transformation is a contraction, if $\mu \geq 1$ then it is a dilation. In the case of curvor chains 'χ' should only be applied to vector component of '**X**' since angle 'χ' is invariant to μ. For non-uniform changes (eg an increase in length but not in breadth) a matrix operator Ξ may be applied to 'V':

$$\Xi(V) = \begin{bmatrix} \xi_1 & 0 \\ 0 & \xi_2 \end{bmatrix} \begin{pmatrix} (v_1)_x, (v_2)_x & \cdots & (v_n)_x \\ (v_1)_y, (v_2)_y & \cdots & (v_n)_y \end{pmatrix} \tag{38}$$

where ξ_1 is the horizontal scaling factor and ξ_2 is the vertical scaling factor - this is a special type of linear distortion found in fairground mirrors.

5b ii TRANSLATIONS

A vector chain of position vectors 'V_p' may either be located in a matrix, or given a new set of axes, by the operation 'Θ' on V_p:

$$\Theta_r(V_p) = \left\{ v_{p1} + p,\ v_{p2} + p \ \cdots\ v_{pn} + p \right\} \tag{39}$$

where 'p' is a vector position (x,y) in the matrix and may for instance be the first recorded absolute address of a vector chain V_R mentioned in section 4a.

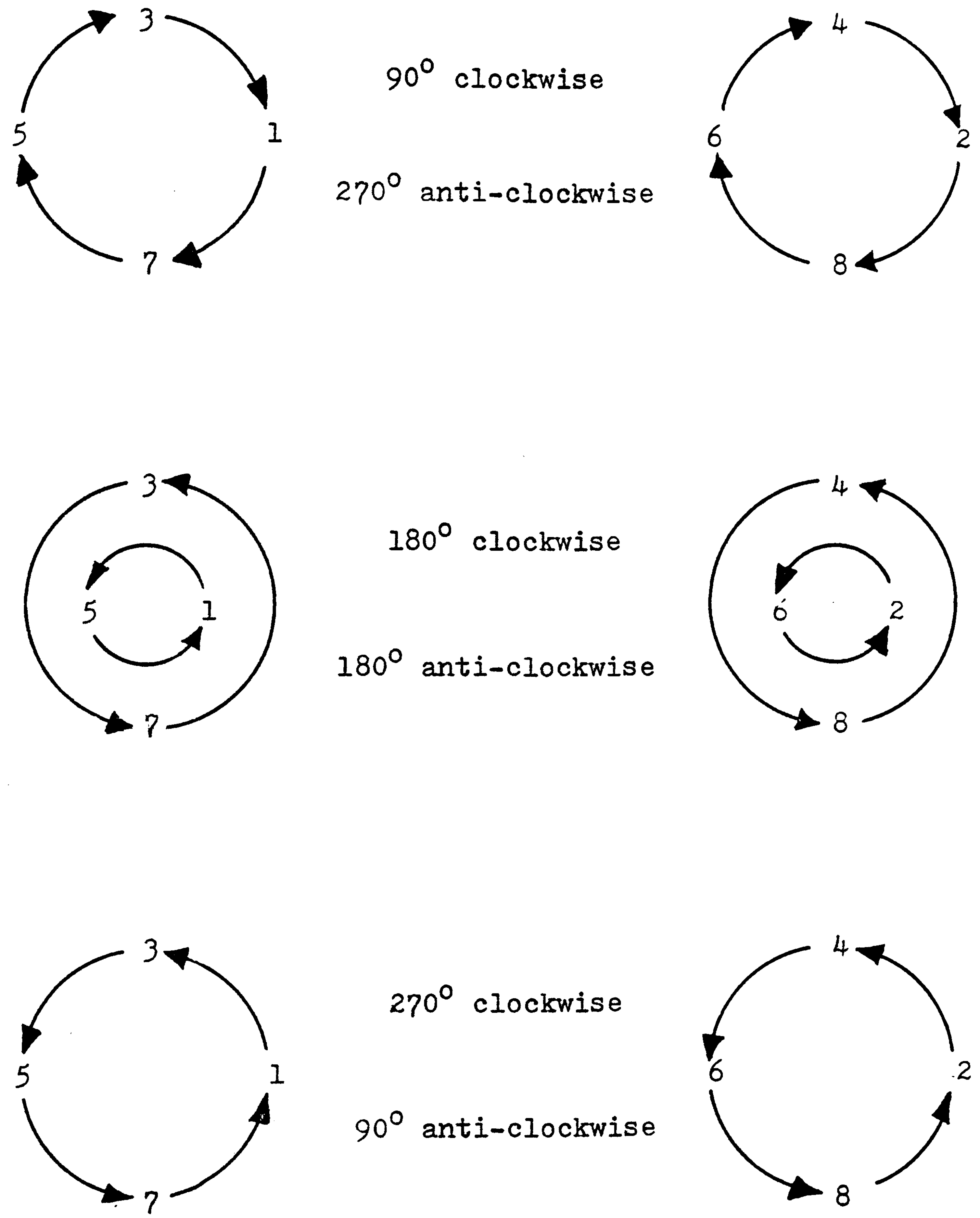
A curvor chain may similarly be located or relocated by adding 'p' to the vector components of 'X' and leaving 'κ' unaffected, as it is invariant to translation, ('X_p' has a vector chain component V_p).

5b iii ROTATIONS

To rotate either the vector chain or its reference frame by angle '$\emptyset$' then a matrix operator '$R_\emptyset$' can act on 'V_p':

$$R_\emptyset(V_p) = \begin{bmatrix} \cos\emptyset & -\sin\emptyset \\ \sin\emptyset & \cos\emptyset \end{bmatrix} \begin{pmatrix} (v_{p1})_x (v_{p2})_x & \cdots & (v_{pn})_x \\ (v_{p1})_y (v_{p2})_y & \cdots & (v_{pn})_y \end{pmatrix} \tag{40}$$

Notice in particular that the rotation is about the "origin" - that is, the beginning point of the chain; other rotations about different initial vectors can be effected by performing the cyclic interchange of expression (27) and integrating as in expression (29), then finally apply $R_\emptyset$; alternatively 'V_p' may be translated by 'Θ_r' and then 'p' becomes the point of rotation. Simple rotations through 0^0, 90^0, 180^0



ROTATIONS

FIGURE 9

and 270^{0} can be achieved on vector state chain (see section 4a) by cyclic replacement of the states - see Figure 9.

For a curvor chain R_{ϕ} may be rewritten as:

$$R_{\phi} = \begin{bmatrix} \cos\phi & -\sin\phi & 0 \\ \sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad (41)$$

thus indicating 'χ' is invariant to rotation.

5b iv PROJECTIONS (STEREOSCOPIC PERSPECTIVES)

To illustrate the concept of projections, consider the human eye. The external world is projected through the iris and the lens to the planar light receptive retina. Thus a three-dimensional scene is translated into a two-dimensional image. In order to see the third dimension (ie depth) we have to have two eyes and correlate the slightly different images to give us range information. Let us now discuss, as promised earlier, binocular vision in terms of simplified point projections.

In Figure 10 two pin hole 'eyes' are positioned a distance 'd' apart and a perpendicular distance 'r' from the retinal plane (x,y,0). A point (x_1, y_1, z_1) is projected through these two holes so as to give rise to two other points $(x', y', 0)$ and $(x'', y'', 0)$ on the retinal plane, as is indicated. Given this geometry it can simply be shown that:

$$\begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix} = \frac{1}{d+x''-x'} \begin{bmatrix} d/2 + x'' & 0 & d/2 - x' \\ y' & d+x''-x' & -y' \\ -r & 0 & r \end{bmatrix} \cdot \begin{bmatrix} x' \\ y' \\ x'' \end{bmatrix} \qquad (42)$$

which directly relates $(x_1\ y_1\ z_1)$ to $(x', y', 0)$, $(x'', y'', 0)$, 'd' and 'r' where $(x_1\ y_1\ z_1)$ is of course the unknown point in space.

Clearly a pair of left and right vector chains - V_{pl} and V_{pr} - can be transformed in this way with rotations and translations of the retinal plane, incorporated as in the previous section, but note that V_{pl} and V_{pr} must be referenced to a common point and confusion between associated elements is conceivable.

The projections of curvor chains is beyond the scope of this text, however it is prudent to remark the angle of curvature '$\varkappa$' is not invariant under this type of projection. This is fairly obvious, since a 'can-top' changes its curvature as you change your own perspective or it is rotated and/or translated. However, prime reference points (eg sharp corners) should be applicable to transformation (42) in order to provide depth information.

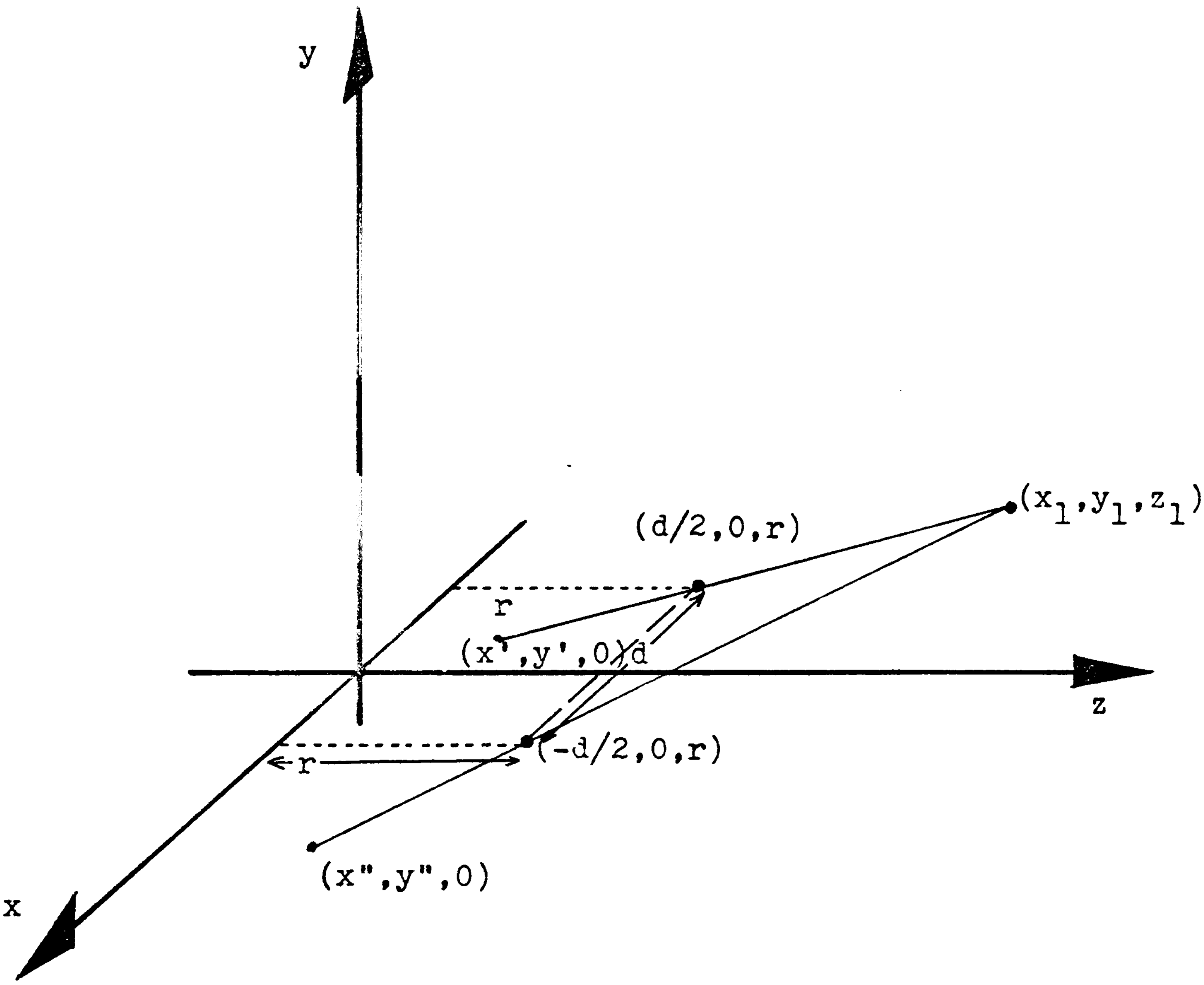
It is also necessary to comment that single projections through a pinhole can give rise to the linear contrations, dilations and translations of a planar object, similar to those described in the previous subsections for vector chains provided the pinhole is positioned between parallel object and image planes - by rotation of the image plane, within its own plane, rotation can be achieved. However, such effects as foreshortening and occlusion arise due to the objects being three-dimensional and this information can only be conclusively established if either the pinhole is allowed to move (ie a change in perspective) or two or more pinholes are used.

5c UNIVERSALITIES

Previous sections 5a and 5b have established certain mathematical properties of vector and curvor chains that reflect the physical universe they describe. Let us now take this one stage further to discuss operations on sets of vector and curvor chains to determine such universal properties (ie properties common to a universal set) as:

1 Containment
2 Congruence and Similarity
3 Concentricity
and
4 Symmetry

the first three of which are mutually referential, the fourth both mutually and self referential.



STEREOSCOPIC PROJECTIONS

FIGURE 10

5c i CONTAINMENT

In order to establish that a vector chain V_a is contained within another vector chain V_b, it is only necessary to establish that a single position vector of V_a is within the confine of V_b (NB both chains must be referenced to a common point or absolute grid), since naturally collected vector chains cannot cross. So as to save time in performing this check, it is useful to first test that:

$$\max\left[V_{pa}\right] \leq \max\left[V_{pb}\right]$$

and

$$\min\left[V_{pa}\right] \geq \min\left[V_{pb}\right] \qquad (43)$$

This, at least, ensures the two chains are within the same neighbourhood and the box parameters of V_a are smaller than those of V_b - note that the vector inequalities must be satisfied over both 'x' and 'y' components.

The premise by which a point is tested to lie within a closed contour (eg a vector position chain) is:

A point is internal to any closed contour if and only if while cycling once around the contour any radial line from the point is crossed an odd number of times.

In practice the containment of one chain within another may be tested for by the following algorithm:

1 Choose any element of a possible inner vector position chain V_{pa} - having first established the box parameter criterion (just outlined) is satisfied.

2 Adjust the position vectors for the possible outer vector position chain V_{pb} so that they are referenced to a point chosen on the inner chain and then choose for convenience the +ve x-axis as the radial line.

3 If $(v_{pb,r})_y$ and $(v_{pb,r+1})_y$ are of different sign and the x-intercept is positive, that is

$$\frac{(v_{pb,r})_y \ (v_{pb,r+1})_x - (v_{pb,r})_x \ (v_{pb,r+1})_y}{(v_{pb,r})_y - (v_{pb,r+1})_y} > 0 \tag{44}$$

then this constitutes a crossing 'c'.

4 If 'c' is set to '1' on the first crossing, and alternates between '0' and '1' on subsequent crossings, then, if c is '1' after a complete cycle, the chain V_a is contained in V_b.

there are of course provisos on this algorithm that relate to the degree of association between vector elements in establishing V_a and V_b but if there is any doubt two or more points, maximal separated, from the inner chain may be used or alternatively maximum length chains may be used (ie vector state chains).

Thus ultimately any set of vector chains may be divided into sub-sets of chains within chains, represented by a containment operator 'con' on C(D), that is:

$$\begin{aligned} \text{con } (C(D)) &= \Big\{ \{V_{1,1}, V_{2,1} \cdots V_{n1,1}\} \ \{V_{1,2}, V_{2,2} \cdots V_{n2,2}\} \\ &\qquad \cdots \ \{V_{1,N} \cdots V_{nN,N}\} \Big\} \\ &= \Big\{ V_{s,1}, V_{s,2} \cdots V_{s,N} \Big\} \end{aligned} \tag{45}$$

where the threshold operator 'B_θ' has been applied to 'M' for many values of 'θ' and each sub-set V_{sR} relates to a set of contours working inwards left to right, for example, as may come about by varying degrees of light intensity.

The containment of a curvor chain within a curvor chain is best dealt with before conversion to a curvor chain has taken place - that is in the vector chain form.

5c ii SIMILARITY AND CONGRUENCE

A chain Φ_a is perfectly similar to Φ_b if and only if:

$$\Phi_a = R_{\not{0}} (\Theta (\mu (\Phi_b))) \qquad (46)$$

is true where $R_{\not{0}}$, Θ and μ are to be determined.

Notice Φ_a and Φ_b may be either relative to each other or an absolute frame (ie position vectors). The translation operator is made to perform the cyclic interchange of the point of rotation as described by expression 27, and must be tried for all 'n_b' members of Φ_b. The scaling operator μ may be determined by a maximisation operator 'MAX' on Φ_a and Φ_b where:

$$MAX(\Phi) = (\text{maximum } v_{pr} \text{ of } V_p \text{ of } \Phi) \qquad (47)$$

and

$$\mu = MAX (\Phi_a)/MAX (\Phi_b) \qquad (48)$$

The rotation of operator $R_{\not{0}}$ is made to perform all rotation through $\not{0}$ ($-180 < \not{0} \leq 180$). If $\mu = 1$, then Φ_a is congruent with Φ_b.

Clearly a considerable amount of work is required in scanning $R_{\not{0}}$ and μ to recognise perfect similarity or congruence of two chains, that can only occur in the number of elements n_a in Φ_a equals the number of elements n_b in Φ_b - this being a useful pre-requisite condition. However, perfect similarity and congruence are rare occurrences in nature and in subsequent sections tolerances on the similarity and congruence of Φ_a and Φ_b shall be introduced.

5c iii CONCENTRICITY

Concentricity is a special combination of containment and similarity, such that if two chains Φ_a and Φ_b form:

$$\Psi_s = \left\{\Phi_a, \Phi_b\right\} \qquad (49)$$

where Ψ_s is a set of contained chains and the relation

$$\Phi_a = \theta(\mu(\Phi_b)) \tag{50}$$

holds then Φ_a is concentric with Φ_b.

Clearly containment can be tested for as in sub-section 5c i, and may be determined as in sub-section 5c ii, or by establishing the ratio of box parameters. As with congruence, the problem of tolerance shall be returned to later.

5c iv SYMMETRY

Symmetry exists in the two major forms of self and mutual symmetry when applied to chains and sets of chains respectively. These may both be further divided into reflexion (of local and global extent) and rotation symmetries.

For a single vector chain 'V' to be rotation symmetric

$$V = R_{\phi}(\theta(V)) \tag{51}$$

should be true. If the above expression is true for 'n' different values of 'ϕ' then this is an n-fold symmetry. θ, the reader is reminde performs cyclic interchanges of elements of V.

For a pair of vector chains 'V_a' and 'V_b' to be rotation symmetric:

$$V_a = R_{\phi}(\theta(V_b)) \tag{52}$$

should be true.

For a set of vector chains to be rotation symmetric:

$$V_a = R_{\phi}(\theta_1(V_b)) = R_{2\phi}(\theta_2(V_c) = \ldots = R_{(N-1)\phi}(\theta_{N-1}(V_r)) \tag{53}$$

should be true for N-1 cyclic permutations of the set, where θ_1, θ_2 ... θ_{N-1} perform different cyclic interchanges of different members of the set. To establish such a relation, proximity tests can first be executed with respect to the chains' centres of gravities, which should of course be equally spaced. The common centre of gravity is the point of rotation or centre of symmetry if expression 53 is true.

For curvor chains the conditions must be extended to include concurrences of the angles of curvature associated with the vector components.

In the case of reflexion symmetries R_ϕ has to be replaced by R'_ϕ, where

$$R'_\phi = \begin{pmatrix} \cos\phi & -\sin\phi & 0 \\ -\sin\phi & -\cos\phi & 0 \\ 0 & 0 & -1 \end{pmatrix} \qquad (54)$$

where for a single curvor chain 'Φ'

$$\Phi_{m-r} = R'_\phi\,(\Phi_{m+r+s}) \text{ for all } r<n/2 + s \text{ and indicial operations modulo } n \qquad (55)$$

where m is the index of symmetry and 's' is a Boolean variable depending on whether the number of elements 'n'' in the chain (or sub-chain) is even (s = 0) or odd (s = 1). If n' is less than n, then this is <u>local</u> symmetry, and if n' is equal to n, then this is <u>global</u> symmetry.

Notice that line symmetry demands the bifurcation of the set into two equal sub-sets - these sub-sets need not be consecutive if the condition of local symmetry is relaxed, or redefined for <u>partial</u> local symmetry.

The values of 'ϕ' for reflexion symmetry may be calculated by first determining the centre of gravity of the considered points 'v_g' and the centre of gravity of the two initial points 'v_{gi}'. It then follows that:

$$\phi = \angle(v_g - v_{gi}), 0 \qquad (56)$$

n-fold reflexion symmetry occurs when expression 55 is true, for 'n' values of 'm' and n = n'.

For a set of curvor chains reflexion symmetry may be defined by:

$$\Phi_a = R'_{\phi}(\Theta_1(\Phi_b))$$

$$\Phi_c = R'_{\phi}(\Theta_2(\Phi_d))$$

$$\Phi_r = R'_{\phi}(\Theta_{\frac{N}{2}}(\Phi_s)) \qquad (57)$$

being entirely true and provided the axes of symmetries are co-linear.

Clearly a deductive approach has been taken here, first introducing containment, then congruence, then concentricity, then symmetry - an inductive approach may have been more satisfactory and aesthetic, since symmetry appears most fundamental.

5d TOLERANCE TO ERRORS

It should finally be stated in this section that error signals can be defined for the definitions of similarity just given of the form:

$$E = Q'(S - T(S')) \qquad (58)$$

where E is an error signal, S is a target set, 'T' is a transformation of S' - the input set, and Q' is a distance metric.

For example, an error in rotation symmetry 'E_s' for a chain 'χ' may be expressed as:

$$E_s = \sum_{r=0}^{r=n} \left| \mathbf{X}_r - R_{\phi}(\mathbf{X}_{r+s}) \right| \quad \text{for all } 0 \leq s \leq n \qquad (59)$$

where '$\mathbf{X}_r$' is a member of 'χ' and r+s is of modulo n.

Clearly in ideal conditions $E_s = 0$, however the universe is not ideal and tolerances have to be set through experience and consistency criteria. The choice of distance metric is purely one of computational or analytic convenience.

The problem of comparing chains of different length can only simply be solved by further association criteria, given the information that the chains are similar - this problem will therefore be returned to in the subsequent sections on pattern languages.

6 VOCABULARIES AND GRAMMARS

The English language allows for the description of most things in minute detail, but to describe a single shape in terms of words can be a most tedious task, a complicated scene - almost impossible. Nevertheless the ability to translate an image into a verbal, written or mental collection of words is there and if the pattern recognition problem is to be solved in general the machine must be given a similar capability. It is Man's gift to manipulate words and reason that has separated him from the rest of the animal kingdom. Words are the fabric of his thought and images his inspiration.

However, I am not suggesting that the 'words' used by the machine should take exactly the same form, only that they should be translatable if required to an appropriate natural language to facilitate communication between the Machine and Man.

In order to reduce the information rate, words have evolved that represent complex shapes (eg a cat, a pen and a leaf), but in point of fact we never actually see such a thing as 'a cat' - we see a collection of features, facets of behaviour and other properties that are consistent with the name 'cat', every cat is unique.

To some extent feature extraction has already been discussed - a curvor is a feature of a shape. In this section methods for translating curvors into 'words' shall be introduced and by deductive reasoning these words may be re-expressed as names - that is if a name exists in memory.

The other beauty of words is that they can be as precise or as vague as is required and this fact also shall be made use of.

The grammatical structure of a set of words comes about through the order imposed by the conversion processes and any variation can only be due to a variation in the image or an equivalence of words in the language. This variation must also be taken into account.

Two 'ad hoc' vocabularies and grammars shall be discussed, one of which is self referential and the other mutually referential. There is a third sort that is relative to an absolute grid, but as this has been well discussed earlier in the paper, it shall not be explicitly included in this part of the text.

6a A SELF REFERENTIAL VOCABULARY AND GRAMMAR

A self referential language is one that is only applicable to a single shape. It is entirely designed to describe a form irrespective of its surroundings. It is therefore independent of orientation, absolute size and translation.

A curvor chain 'χ' can easily be translated into such a language. The first step is to rewrite each curvor component 'X_r' as a 3-tuple 'F_r', that is

$$F_r = \langle \text{percentage length, percentage curvature, percentage sharpness} \rangle$$
$$= \langle \%l_r, \%c_r, \%sh_r \rangle \qquad (60)$$

The percentage length is calculated by finding the curvor's length and dividing by the total perimeter of the shape (using the techniques of Section 5a iv). The percentage curvature is found by taking the angle of curvature of a curvor and dividing this by the sum of all the absolute angles of curvature for the complete chain and the percentage sharpness is found by taking the angle between the rth and the (r+1)th curvors defined by

$$sh_r = \angle v_r, v_{r+1} - (\varkappa_r + \varkappa_{r+1}) \qquad (61)$$

where 'sh_r' is the <u>sharpness</u> and dividing this by the sum of all the absolute sharpnesses for the chain. The last member of chain is reference to the first and multiplication by 100 to give a percentage is arbitrary, that is other units may be used. These processes are usually called 'normalisation'. Thus a square is represented by a feature chain ' $\mathcal{F}$ ' as:

$$\mathcal{F} = \left\{\langle 25, 0, 25\rangle, \langle 25, 0, 25\rangle, \langle 25, 0, 25\rangle, \langle 25, 0, 25\rangle\right\} \tag{62}$$

which may be translated as a language ' $\mathcal{L}$ ' as:

$\mathcal{L}$ = 4 straight lines of equal length and at equal angles to each other (63)

Notice that in natural language there are some irrelevant words but this is only due to the immense generality of our language. Clearly it would not be difficult to define tests to translate between (62) and (63) but this will have to be left mainly to the reader. The point to notice is that of 'significance' - normally a feature is significant if it is either large, sharply discontinuous or very rounded, all of which can be rapidly tested for. In the case of the letter 'A' used in previous sub-sections, the main features are its long sides, its pointed (or rounded) 'top' and its two sharp 'feet', again all of which can be rapidly extracted (NB top and feet do not belong to a self referential language). A typical set of words to describe a shape in a self referential way is given below in Table 4.

NOUN	ADJECTIVES
line	1, convex/straight/concave, slightly, very, least,most shorter/est, longer/est left, right, above, below
lines	2, 3, ...n, not, equal, parallel shorter/longer sharper/est, blunter/est left, right, above, below
angle	1, smaller/est, larger/est zero, 45^{o} right
angles	2, 3, ...n, not, equal, zero, smaller/larger, 45^{o} right
shape	not, curvacious, angular, symmetric short, long, square/round

+ verbs, articles, prepositions and conjunctions

TABLE 4

There is obviously much commoness of meaning in such a table, but we do not want to invent boring machines. In fact all equivalences, such as 'long' and 'not short' must be pre-programmed and random selections made if required.

The usages of a self referential feature chain are not great in the form described (eg automatic analysis of chromosomes and simple geometric shapes) but with some modifications and extensions, such as a partial reference to an absolute grid, they have many applications, the most important being character recognition.

A typical list of questions to derive a set of useful properties from a chain in the terms of Table 4 may be as follows:

General Questions

1 What is the perimeter?
2 What is the absolute curvature?
3 What is the absolute sharpness?
4 What are the shape's box parameters?
5 Is the shape globally symmetric?
6 How many features are there?

Particular Questions

1 What is the main feature?
2 Are any features repeated? (actually many questions)
3 Where are the longest lines?
4 Where are the sharpest angles?
5 Where are the most convex/concave lines?
6 How are the features inter-related?

Clearly all these questions may be answered by techniques previously described and structured in a way imposed by the order of the questions asked as a deductive description. Each answer prompts a fresh question until a minimum description is obtained to precisely describe the shape. Notice that basic grammatical rules of natural language can be imposed, such as adjectives preceding nouns etc, and the theories of syntactic analysis applied. Also the problem of comparing one chain with another of different length has disappeared since only properties questioned are searched for, anything that remains must be noise or unnecessary detail in describing the shape or possibly suggesting that the shape under scrutiny does not fit in with the shapes already scrutinised.

6b A MUTUALLY REFERENTIAL VOCABULARY AND GRAMMAR

A mutually referential language is a special extension of a self referential language. The main difference is that it can deal with a set of chains that inter-relate and is thus far more powerful.

The first requirement is that each chain has associated with itself a set of parameters that specify for example:

a its size (ie its box parameters)
b its orientation (ie the angle of its longest dimension
c its overall curvature
d its overall sharpness
e its relative position

and that these be normalised over all images in the set. Clearly this will form a set of n-tuples 'Γ' that specify of overall properties of the set of chains.

By using these techniques it is possible to increase Table 4 to produce the mutually referential vocabulary of Table 5 given below:

NOUNS	ADJECTIVES
line	1, convex/straight/concave, slightly, very, least, most shorter/est , longer/est left, right, above, below
lines	2, 3, ...n, not, equal, parallel shorter/longer sharper/est, blunter/est left, right, above, below
angle	1, smaller/est, larger/est zero, 45^{o} right
angles	2, 3, ...n, not, equal, zero, smaller/larger, 45^{o} right
shape	not, curvacious, angular, symmetric short, long, square/round

NOUNS	ADJECTIVES
~~Shapes~~ Shape	1, 2, 3, 4 n
	not, inner/outer
	small/large
	near/far
	left/right, above/below
	most/least
	simple/complex
	concentric
	congruent
	symmetric

+ verbs, articles, prepositions and conjunctions

TABLE 5

Notice that the techniques already outlined can easily define these words.

The set of questions to produce the complete description of a set of chains can be the following extension of these for the self referential language:

General Questions

1 How many shapes are there?
2 Are they simple or complex?
3 Are they symmetrically placed?
4 Are they congruent (similar)?

Particular Questions

1 How many shapes are congruent?
2 How many shapes are concentric?
3 How many shapes are symmetric (rotation, reflexion)?
4 Do any shapes form clusters?

Of course the questions need to be supplemented for particular classes of pattern, but they are fairly typical of the basic questions asked when faced with an unknown image - that is, a breakdown into the known.

In the next section the problems of machines learning to recognise a shape and sets of shapes shall be discussed in terms of learning a language through conversation.

7 LEARNING A LANGUAGE

Suppose that the vocabulary and grammar of the previous section is formalised in the following way.

Let there be a set of N general questions Q_g, where:

$$Q_g = \left\{Q_{g1} \ldots Q_{gN}\right\} \quad (64)$$

implicit in a recognition machine (to be described) and let there also be a set R_g of N general responses determinable from the input matrix, that is:

$$R_g = \left\{R_{g1} \ldots R_{gN}\right\} \quad (65)$$

These responses in fact form an address in N-space, if the responses are given a numerical code.

At this address, let there be sufficient memory for a name (eg drawing) which is given to the machine by an external source if there is no name at the address, (alternatively if there is no external source, then the machine may invent a name - eg an unused number). This name may be requested by the machine from the external source. This is the beginning of a conversation, and the machine has already extended its inbuilt vocabulary by one. Notice that the address given by R_g shall be unique if the questions are sufficiently general - if a similar name has to be given twice either the name is insufficiently precise, or the response insufficiently general.

Now the name of the class may be all that is required of the machine in order for it to act (eg return the name or avoid a collision), in which case the input matrix can be re-assessed and, if the same, new information requested or sought. Thus a simple recursion through the set of questions Q_g is set up. However, it may be that the name is insufficient to act upon and further information must be gathered. In this case a set of 'n' particular questions 'Q_p' must be asked, that is:

$$Q_p = \left\{Q_{p1} \ldots Q_{pn}\right\} \quad (66)$$

which will produce a set of 'n' responses R_p, where:

$$R_p = \left\{R_{p1} \ldots R_{pn}\right\} \tag{67}$$

Now the order and number of questions in Q_p must initially be general, and thus Q_g and Q_p are initially similarly structured; however, by monitoring the usefulness of a particular response in the context of circumstances of the machine, Q_p may be ordered such that:

$$Q_{pr} = \underline{R}(R_{pr-1}) \tag{68}$$

where this simply means that the next question is a function 'F' of the previous question's response. This deductive process is essentially no different from the next address given after the set of general questions which produced a class name, except the internal structure of Q_g is:

$$Q_{gr} \neq F(R_{gr-1}) \tag{69}$$

In other words $\left\{Q_g, R_g\right\}$ is an <u>inconsequential</u> question/response set and $\left\{Q_p, R_p\right\}$ is a <u>consequential</u> question/response set.

Now the automatic determination of 'F' for a consequential set is no simple matter, but the labelling (naming) is of course executed in the same way as for the inconsequential set, with a conversation ideally taking place between the external source (trainer) and the machine (the learner). Notice that the final name of a consequential set must have associated with it an address to return to the general and a method to ensure no continuous cycling around the same path - thus all learning must be unrepetitive.

It should be clear that a response in the case of pattern recognition is normally a name and that a name generates a whole set of further questions (addresses) until a terminal point is reached. Thus implicit in this technique is a theory of motivation (goal direction - the will to learn) and reasoning ability. The implicit capacity to deduce the specific from the general is of course there, in that the order of the questions narrows down the number of possibilities, and in contrast, the implicit capacity to induce the general from the specific is also included, since individual stimulii are related to the universal properties.

The important point is that if a machine is given a close coupling with Man and his environment through language, it can be made to appear to possess:

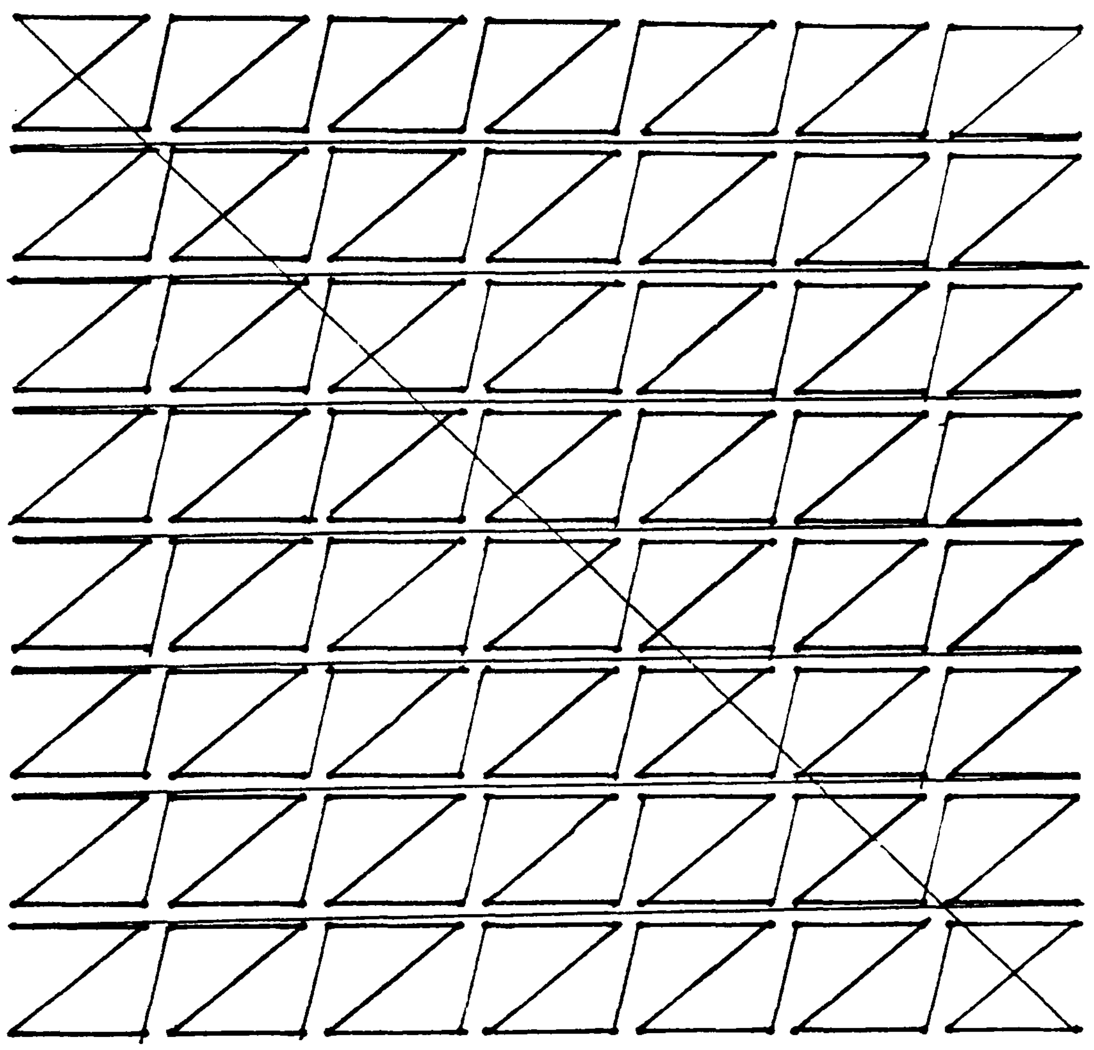
1 Curiosity
2 Intelligence
3 Motivation
4 Ability to learn
5 Eloquence

and many other desirable traits.

In fact once a machine has been taught, it can become the trainer and train either other machines or humans. This may be better achieved by being able to perform the opposite to pattern recognition, that is pattern generation, by outlining the steps back from the final word (or name). This is achieved by retracing the deductive steps and at each question and answer node generating the appropriate name, thus forming an inductive statement to help teach the other person/machine what is meant by the word in relation to the general. Clearly this form of teaching may well be quicker since the number of questions will only be limited to those necessary.

8 THE CONSTRUCTION OF A SEEING MACHINE

In the previous sections the mathematical structure of a "seeing machine" has been suggested. In this section, by way of summary, the construction of such an artifact will be discussed. As a criterion of excellence the performance of the human visual system shall be used, as this will ensure at least a compatible man-machine match, in that the man will not be waiting for the machine to react. The question of whether a recognition machine can be built (using present technology for a reasonable cost) to out-perform Man in most respects shall also be considered.



Z-TYPE SCANNING

FIGURE 11

8a THE PRE-PROCESSING

There are many ways to gather information in the form of a matrix representation described in Section 2, but perhaps the most familiar and economical is to use a TV camera feeding into a random access memory (RAM) (other parallel methods are arrays of photodiodes or charge-coupled light sensitive devices - the technology from the point of view of cybernetics is largely irrelevant). In order that the TV signal may be rapidly transformed into a logically differenced matrix, the output signal may be thresholded and the scanning beam modulated so as to move in a Z type scanning motion depicted in Figure 11.

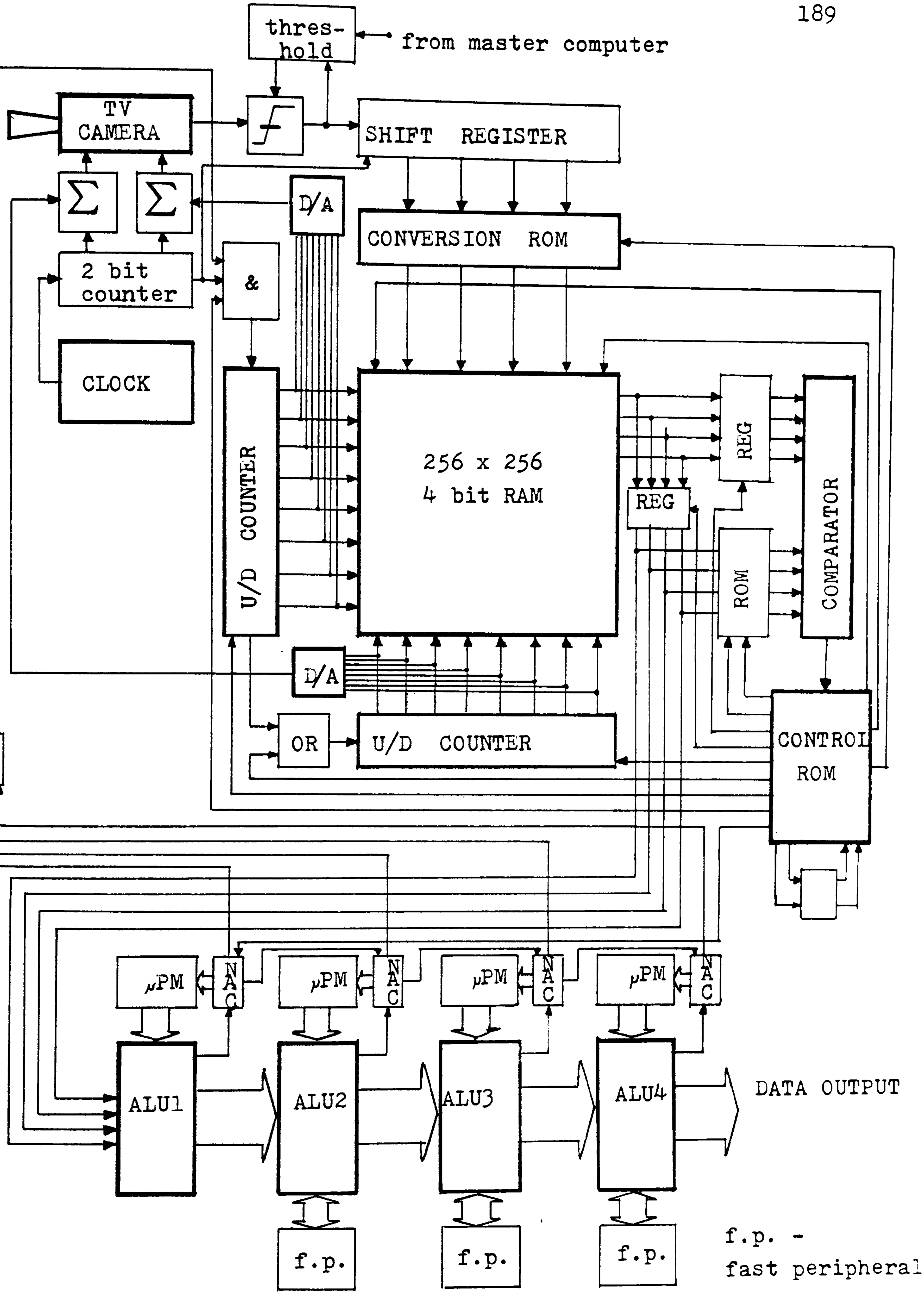
The video signal is them sampled at the four corners of each Z motion and by using a read only memory (ROM) as a "look-up", translated into 256 x 256 x 4 bit RAM as a logically differenced matrix.

Thus in around 1/25 second the image 'seen' by the TV camera has been both binarised and logically differenced (thickening shall not be dealt with here), ready for further processing. By pipelining, this process can be made to take place continuously for there are no problems with switching speeds (typically <50ns for most bipolar gates) or memory access times (<200ns for CMOS memories - not the fastest).

For a colour signal, regions of equivalent code may be tested for and logically differenced. Thus in one second, 25 differently coded colour components of the scene may be logically differenced using this serial type of approach. Obviously with parallel array processing this may be increased by factor in excess of a thousand.

Thus the information in now in RAM, by x,y addressing this information by two up-down counters controlled by the current state, the logically differenced matrix may be translated into a set of vector chains in the way described in Section 4c. Assuming that every one of the 'N' calls in the RAM demands all three possible neighbouring cells to be tested for compatibility, then the number of cells visited is at most 3N, and since N cells must be visited to scan the complete matrix regardless of the image the number of cells actually visited 'n' is such that

$$N \leq n < 3N \qquad (70)$$



A PRE-PROCESSING SYSTEM

FIGURE 12

Therefore provided the RAM is just three times faster than the scanning process and the decision electronics takes a negligible time to operate, then the difference matrix can be completely translated into M vector chains in 1/25 second.

The combined maximum length of all the vector chains is less than N, so that the storage required should be a similar N,4 bit words long where 'N' is 64K in the example already considered.

The state and vector association processes are best dealt with by fast arithmetic logic units (ALUs) under micro-program control and require a wider word (eg 16 bits) to function efficiently.

Supposing that the state association process takes on average 5 machine cycles, with a cycle time of 150ns, then for the most complex of images the total time taken is less than 1/25 second. Similarly, vector association may be achieved, using the same electronics with a special purpose angle calculator with hardware multiplier added in and inverse cosine look-up table, in an equivalent or shorter time, as the chain has fewer members. Clearly curvor association and sharpness calculations require the same electronics and should operate on similar or shorter time scales.

The just-described pre-processing system is shown in Figure 12, but instead of recording the intermediate states of the association processes, the ALUs can stop the clock accessing the RAM whilst they catch up in the unlikely event of their small buffer stores becoming full. Since the association processes increase in complexity but reduce the number of elements in the chain, a balance should be achievable.

Finally in this sub-section, it should be remarked that this process could be made much quicker by greater duplication of circuitry and the stitching together of vector chains collected over smaller input matrices.

8b THE PROCESSING

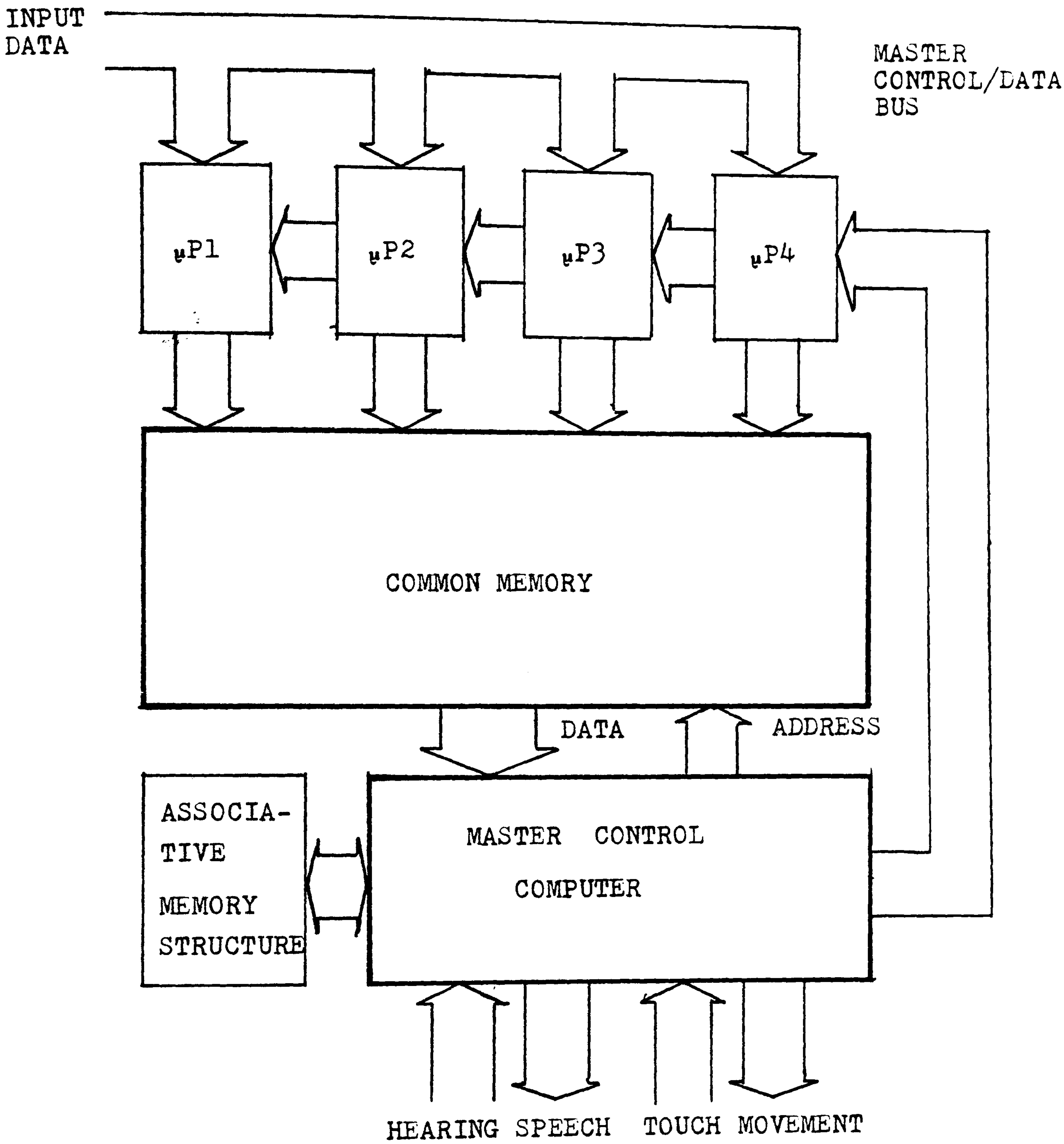
The calculations of such quantities as the area enclosed by a vector chain or the perimeter of a curvor chain can either be carried out by storing up the chains in memory and calculating these quantities sequentially by a single processor, or by feeding the appropriate chains in parallel to many processors on a common bus - so as not to slow the processing down.

The algorithms followed shall depend on what is required of the system and cannot be definitively laid down without this knowledge. However, if the recognition machine is to possess similar capabilities to the human visual system, it must be able to cope with all the geometric properties of Section 5 such as containment and congruence.

The ideal device for performing these tasks is the microprocessor with a small amount of inbuilt ROM and RAM for control and data manipulation. The control of these microprocessors would come from the post processing unit or learning section of the electronic system. The results of demanded calculations being placed in an allotted area of common store accessible by the post processing unit. The file structure and extents of these intermediate areas requires further research, and is of both self and mutually referential type.

It should be noted that the algorithms are still precisely defined within the confines of limiting parameters and would present no major problem to translate into software. Their speed of operation should be well within the 1/25 second proposed for the pre-processing if a parallel approach is adopted. This means that the ability to follow moving objects should be quite within the capabilities of this machine; however, this would require a special processor to check congruence between analysed frames.

It may be desirable to have an interrupt system such that if a very large and significant shape came along, work on other shapes could be stopped, and analysis of this shape begun immediately. A priority system of this sort would make an interesting study and would only be possible if the processing system could provide some global information almost immediately, such as a chain's box parameters.



A PROCESSING AND POST PROCESSING SYSTEM

FIGURE 13

It should perhaps have been mentioned earlier that this is the most appropriate unit to stitch together curvor chains if the TV camera is allowed to traverse or zoom in or out on a scene.

8c THE POST PROCESSING

The assignment of a name or group of names and the communication and learning of language is best achieved by the final controlling element in the system. This controlling element has to possess the ability to monitor and control many elements (peripherals) simultaneously and have an inherent flexibility. The only choice here, within our present technology, is a large-scale general purpose computer with some associative memory and separate inputs from other peripheral processing devices (eg speech recognition and speech synthesis system) as is shown in Figure 13 along with the processing of the last subsection.

The architecture of this system needs no description except perhaps the associative memory, which is essentially 'contents addressable' and may be either hardware or software implemented.

The problems of the post processing section are largely software, as opposed to hardware. There is a need to have '<u>self generating</u>' programmes that can construct pathways through the associative memory that may be traversed in either direction (inductive/deductive logic). This suggests a string orientated language with end points that are link words (actually addresses) with other strings, a string either being a command (or question) to the processing section or an output comment or action - the translation between machine language and English (natural language) taking place within this network.

Clearly if the associative memory is not to grow unwieldy, it is necessary to structure extension stores that may be of a slower, serially oriented type, such as bubble memory. This store could be designed so that information that is not being constantly used is slipped further and further back, by indirect addressing into this memory until ultimately it is overwritten to make room for more significant data - forgetting is a very important process.

Most high level programmes cannot change their written structure. This is mainly because of their totally numerical bases and exact requirements. Nevertheless there is no reason why a programming language should not be invented that optimises itself by understanding the nature of its non-numerical data. It is really only a question of rewrite rules and optimality criteria being initialised and then made to feed back on themselves. This is another area for further research.

The overall question of feedback in a recognition system, such as described, has been largely glossed over. For example, it is thought important that all association thresholds should be controlled by the master control computer. However, this has not been explicitly stated elsewhere. It is therefore worth emphasising that without feedback control to the following areas:

1 The lens/lenses and direction of the TV camera
2 The binariser
3 The up-down counters
4 The association processors
5 The microprocessors calculating the scenes' properties (eg symmetry)

the system would be very special purpose and inflexible. Feedback allows for flexibility. The most crucial area where feedback is to be used is in the file structuring of the common memory. Clearly there is no point in creating a file in memory if the master controller does not know of its existence - in which case it must have been provoked to instigate it.

The abilities of such a system in comparison to a human's are almost as great, and with more efficient circuit design, may certainly be made more rapid, with individual memories probably exceeding the capacities of the human brain. It is a daunting thought that machines can now be constructed that can have greater vocabularies, see more precisely and probably survive longer than Man.

9 CONCLUSIONS

What I hope I have achieved in this paper is to have supplied the reader with the necessary and sufficient information to be able to understand the design and construction of a "seeing machine" within the confines of our technology. In no way has there been an attempt to draw analogies with the human visual system except in terms of its input and output - for this type of approach the reader is referred to Leonard Uhr's excellent papers on such a system called SEER (see reference).

Neither has the conventional decision / theoretic or syntactic methods - been entirely adopted in my approach, for both methods lack in generality and are normally only applied to separate a few classes of pattern and use large training sets although it is certain that each step in the described recognition scheme could be fitted into such an 'ad hoc' hierarchy.

My aim has been, I hope, that of Wiener's, to use the general or universal properties of all sets to distinguish forms. But what of Wiener's integral given in the introduction to this paper? The distinctions between S, T and Q have not always been that clear. This is because it has been applied both locally and globally, but essentially the order of the sections follows the order of defining in general 'S', 'T' and 'Q' for a visual stimulus, where 'Q' is in general a questioning operator and 'T' is a relation between members of 'S'. The averaging with respect to 'T' ensures all possibilities have been considered. By various techniques of expressing components relatively, the need to perform this averaging has been avoided, thus speeding up the process.

It is also somewhat odd that Wiener did not mention the role of language in learning to see, but perhaps this is not too surprising when most people thought all human faculties totally beyond the machine before cybernetics was fathered.

However, I do not wish to claim great insight into the relationship between natural languages and pattern recognition for the last sections are unavoidably vague and need further clarification. I can only wish that they may provoke thought and development of new computer languages.

10 ACKNOWLEDGMENTS

I should like to acknowledge the joint support of the Electonics Department, Chelsea College, University of London and that of Marconi Avionics Limited, Borehamwood, for the use of their computing facilities in the simulation of the preprocessing part of the system. In particular I thank Dr Haneef Fatmi (Chelsea College) for his continuing encouragement and Andrew Hadden (Marconi Avionics) for his useful advice on the problem of containment of a point.

11 REFERENCES

1 FU, K S, 'Syntactic Methods in Pattern Recognition', Academic Press, 1974

2 FREEMAN, H, 'On the Encoding of Arbitrary Geometric Configurations', IRE Transactions on Electronic Computers, EC-10 No 2, June 1961

3 GOLAY, M J E, 'Hexagonal Parallel Pattern Transformations', IEEE Trans. Computers 18 pp 733-740, 1969

4 UHR, L, 'A Model of Form Perception and Scene Description', Computer Sciences Technical Report No 231, University of Wisconsin - Madison, November 1974

5 ULLMANN, J R, 'Pattern Recognition Techniques', Butterworth, 1973

6 WIENER, N, 'Cybernetics or Control and Communication in the Animal and the Machine', MIT Press, 1948

APPENDIX FOUR

A COMPUTER PROGRAM TO PERFORM MULTIPLE VECTOR CHAIN EXTRACTION FROM A BINARY PATTERN

Foreword

The following computer program was written for the Elliott 503 computer in ALGOL to perform vector chain extraction from a binary matrix, the details of which have been given in Appendix Three. No attempt at optimising the code has been made.

After the program, a typical input matrix is given, together with its logical differenced derivative. The output to this given input is then illustrated pictorially as four vector chains of different degrees of association.

```
comment Contour Follower;
begin integer e, h, i, j, k, l, n, r, s, sw;
real a;
begin read reader (3), h, i, j, a, sw;
begin integer array A[-1:i,-1:i], V[1:2,1:j,1:h], W[1:2,1:h],
                    T[0:1,0:1,0:1,0:1], t[1:8,1:2,1:3],
                    M[1:8,1:2],    m[1:8,1:2],    E[1:h],    L[1:8,1:2];
switch ss:= RETURN, LOOP, MISS, HOP, STEP, JUMP;


comment transition table;


T[0,0,0,0]:=0;  t[1,1,1]:=1;  t[1,1,2]:=5;  t[1,1,3]:=7;  m[1,1]:=1;   M[1,1]:=1;   L[1,1]:=2;
T[0,0,0,1]:=0;  t[2,1,1]:=7;  t[2,1,2]:=3;  t[2,1,3]:=1;  m[2,1]:=1;   M[2,1]:=1;   L[2,1]:=3;
T[0,0,1,0]:=0;  t[3,1,1]:=7;  t[3,1,2]:=3;  t[3,1,3]:=1;  m[3,1]:=1;   M[3,1]:=0;   L[3,1]:=4;
T[0,0,1,1]:=1;  t[4,1,1]:=5;  t[4,1,2]:=1;  t[4,1,3]:=3;  m[4,1]:=-1;  M[4,1]:=-1;  L[4,1]:=5;

T[0,1,0,0]:=0;  t[5,1,1]:=5;  t[5,1,2]:=1;  t[5,1,3]:=3;  m[5,1]:=-1;  M[5,1]:=-1;  L[5,1]:=6;
T[0,1,0,1]:=3;  t[6,1,1]:=3;  t[6,1,2]:=7;  t[6,1,3]:=5;  m[6,1]:=-1;  M[6,1]:=-1;  L[6,1]:=7;
T[0,1,1,0]:=0;  t[7,1,1]:=3;  t[7,1,2]:=7;  t[7,1,3]:=5;  m[7,1]:=-1;  M[7,1]:=0;   L[7,1]:=8;
T[0,1,1,1]:=2;  t[8,1,1]:=1;  t[8,1,2]:=5;  t[8,1,3]:=7;  m[8,1]:=1;   M[8,1]:=1;   L[8,1]:=1;


T[1,0,0,0]:=0;  t[1,2,1]:=2;  t[1,2,2]:=6;  t[1,2,3]:=8;  m[1,2]:=1;   M[1,2]:=0;   L[1,2]:=8;
T[1,0,0,1]:=0;  t[2,2,1]:=8;  t[2,2,2]:=4;  t[2,2,3]:=2;  m[2,2]:=-1;  M[2,2]:=1;   L[2,2]:=1;
T[1,0,1,0]:=7;  t[3,2,1]:=8;  t[3,2,2]:=4;  t[3,2,3]:=2;  m[3,2]:=-1;  M[3,2]:=1;   L[3,2]:=2;
T[1,0,1,1]:=8;  t[4,2,1]:=6;  t[4,2,2]:=2;  t[4,2,3]:=4;  m[4,2]:=-1;  M[4,2]:=1;   L[4,2]:=3;

T[1,1,0,0]:=5;  t[5,2,1]:=6;  t[5,2,2]:=2;  t[5,2,3]:=4;  m[5,2]:=-1;  M[5,2]:=0;   L[5,2]:=4;
T[1,1,0,1]:=4;  t[6,2,1]:=4;  t[6,2,2]:=8;  t[6,2,3]:=6;  m[6,2]:=1;   M[6,2]:=-1;  L[6,2]:=5;
T[1,1,1,0]:=6;  t[7,2,1]:=4;  t[7,2,2]:=8;  t[7,2,3]:=6;  m[7,2]:=1;   M[7,2]:=-1;  L[7,2]:=6;
T[1,1,1,1]:=0;  t[8,2,1]:=2;  t[8,2,2]:=6;  t[8,2,3]:=8;  m[8,2]:=1;   M[8,2]:=-1;  L[8,2]:=7;
```

```
for k:=1 step 1 until i do begin
for l:=1 step 1 until i do begin
read A[k,l]; end; end;

begin print punch (3), ££14??; end;


for k:=1 step 1 until i do begin
for l:=1 step 1 until i do begin
print punch (3), sameline, digits (1),  A[k,l]; end;
print punch (3), ££1??; end;

begin print punch (3), ££14??; end;

comment logical differencing;

for k:=1 step 1 until (i-1) do begin
for l:=1 step 1 until (i-1) do begin
A[k,l]:=T[A[k,l],A[k,l+1],A[k+1,l],A[k+1,l+1]];
print sameline, digits (1),  A[k,l]; end;
print  ££1??; end;

begin print punch (3), ££14??; end;

for k:=-1 step 1 until i do
A[-1,k]:=A[k,-1]:=A[i,k]:=A[k,i]:=0;

comment chain extraction;

e:=0;  r:=1;

for k:=1 step 1 until (i-1) do begin
for l:=1 step 1 until (i-1) do begin
s:=1;
if A[k,l]=0 then goto MISS;
LOOP: e:=e+1;
RETURN: if A[k,l]=s then begin
V[1,e,r]:=s;  A[k,l]:=0;

if A[k+m[s,2],l]=t[s,1,2] then begin k:=k+m[s,2]; s:=t[s,1,2]; goto LOOP; end;
if A[k+m[s,2],l]=t[s,2,2] then begin k:=k+m[s,2]; s:=t[s,2,2]; goto LOOP; end;
if A[k+m[s,2],l+m[s,1]]=t[s,1,3] then
begin k:=k+m[s,2]; l:=l+m[s,1]; s:=t[s,1,3]; goto LOOP; end;
if A[k+m[s,2],l+m[s,1]]=t[s,2,3] then
begin k:=k+m[s,2]; l:=l+m[s,1]; s:=t[s,2,3]; goto LOOP; end;
if A[k,l+m[s,1]]=t[s,1,1] then begin l:=l+m[s,1]; s:=t[s,1,1]; goto LOOP; end;
if A[k,l+m[s,1]]=t[s,2,1] then begin l:=l+m[s,1]; s:=t[s,2,1]; goto LOOP; end;
end;
```

```
if s<3 then begin s:=s+1; goto RETURN end;
V[1,r]:=k; V[2,r]:=1, E[r]:=e; r:=r+1; e:=0;
MISS: end; end;
r:=r-1;

comment state association;

e:=n:=1;
for k:=1 step 1 until r do begin
V[2,1,k]:=1;
for l:=1 step 1 until (E[k]-1) do begin
if V[1,l,k]=V[1,l+1,k] then
begin n:=n+1;  goto JUMP;  end;
e:=e+1; n:=1;
JUMP:  V[1,e,k]:=V[1,l+1,k];  V[2,e,k]:=n;
end;
E[k]:=e;  e:=n:=1;  end;

comment state to vector translation;

e:=1;
for k:=1 step 1 until r do begin
for l:=1 step 1 until E[k] do begin
s:=V[1,l,k];  n:=V[2,l,k];
if (V[1,l+1,k]=L[s,1] or V[1,l+1,k]=L[s,2])
and (V[2,l+1,k]=1 or n=1) and l≠E[k]
then begin
V[1,e,k]:=M[s,1]*n+M[V[1,l+1,k],1]*V[2,l+1,k];
V[2,e,k]:=M[s,2]*n+M[V[1,l+1,k],2]*V[2,l+1,k];
l:=l+1;  goto HOP; end;
V[1,e,k]:=M[s,1]*n;
V[2,e,k]:=M[s,2]*n;
HOP: e:=e+1;  end;
E[k]:=e-1; e:=1; end;

comment vector association;

e:=1;
for k:=1 step 1 until r do begin
for l:=1 step 1 until (E[k]-1) do begin
if a<((V[1,e,k]*V[1,l+1,k]+V[2,e,k]*V[2,l+1,k])/
(V[1,e,k]↑2+V[2,e,k]↑2)↑ .5/(V[1,l+1,k]↑2+V[2,l+1,k]↑2)↑ .5)
or (abs(V[1,e,k])≤1 and abs(V[2,e,k])≤1 and sw=1)
or (abs(V[1,l+1,k])≤1 and abs(V[2,l+1,k])≤1 and sw=1)
then begin
V[1,e,k]:=V[1,e,k]+V[1,l+1,k];
V[2,e,k]:=V[2,e,k]+V[2,l+1,k];
goto STEP; end;
e:=e+1;  V[1,e,k]:=V[1,l+1,k];    V[2,e,k]:=V[2,l+1,k];
STEP:end;
E[k]:=e;  e:=1; end;
```

```
for k:=1 step 1 until r do begin
print punch (3), digits (3), W[2,k], sameline, W[1,k];
for l:=1 step 1 until E[k] do begin
print punch (3), digits (3),V[1,l,k], sameline , V[2,l,k]; end;
print punch (3), ££l2??; end;
end;
end;
end;
```

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	1	1	0	1	1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	1	1	1	0	0	1	1	0	0	0	0	0	0
0	0	0	0	0	0	0	1	1	0	0	0	1	1	0	0	0	0	0	0
0	0	0	0	0	0	1	1	0	0	0	0	1	1	1	0	0	0	0	0
0	0	0	0	0	0	1	1	1	0	0	0	0	1	1	0	0	0	0	0
0	0	0	0	0	0	1	1	0	0	0	0	0	1	1	0	0	0	0	0
0	0	0	0	0	1	1	1	0	0	0	0	0	1	1	1	0	0	0	0
0	0	0	0	0	1	1	1	1	0	0	0	1	1	1	1	1	0	0	0
0	0	0	0	0	1	1	1	1	1	1	0	1	1	1	1	1	0	0	0
0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0
0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0
0	0	0	0	1	1	1	1	0	0	0	0	0	1	1	1	1	0	0	0
0	0	0	0	1	1	1	0	0	0	0	0	0	0	1	1	1	0	0	0
0	0	0	0	1	1	0	0	0	0	0	0	0	0	1	1	1	0	0	0
0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	1	1	1	0	0
0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Typical Binary Input

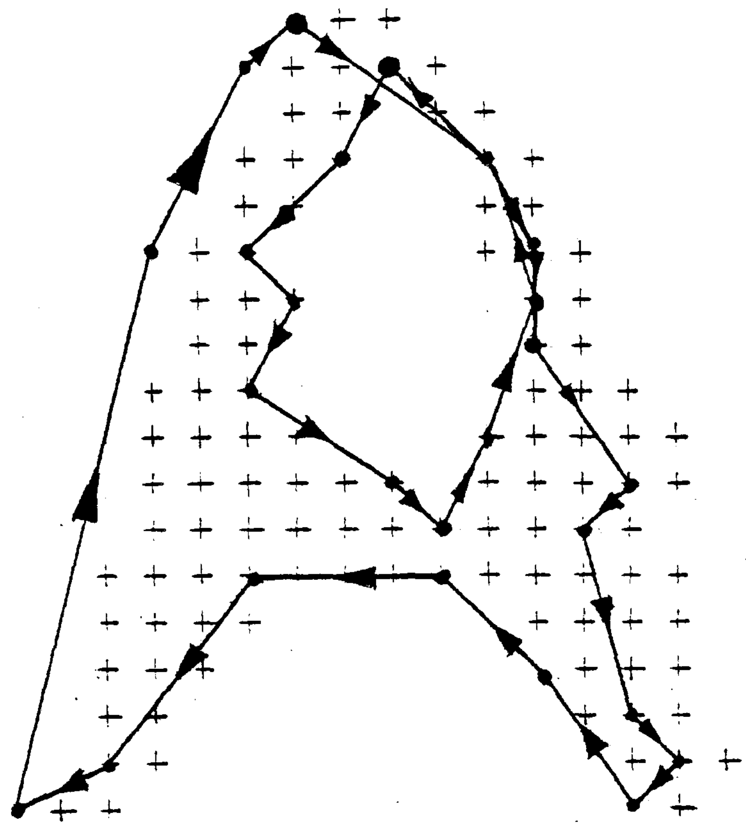
0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	2	0	8	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	3	0	6	4	8	0	0	0	0	0	0	0
0	0	0	0	0	0	0	2	0	7	0	4	8	0	0	0	0	0	0
0	0	0	0	0	0	3	0	6	0	0	3	0	7	0	0	0	0	0
0	0	0	0	0	0	2	6	0	0	0	3	0	8	0	0	0	0	0
0	0	0	0	0	3	0	8	0	0	0	0	4	0	7	0	0	0	0
0	0	0	0	0	3	0	6	0	0	0	0	3	0	7	0	0	0	0
0	0	0	0	0	2	0	7	0	0	0	0	3	0	8	0	0	0	0
0	0	0	0	3	0	0	8	0	0	0	0	2	0	0	8	0	0	0
0	0	0	0	3	0	0	0	8	1	0	3	0	0	0	0	7	0	0
0	0	0	0	3	0	0	0	0	0	8	2	0	0	0	6	0	0	0
0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	8	0	0	0
0	0	0	3	0	0	0	6	5	5	5	5	4	0	0	0	7	0	0
0	0	0	3	0	0	6	0	0	0	0	0	0	4	0	0	7	0	0
0	0	0	3	0	6	0	0	0	0	0	0	0	3	0	0	7	0	0
0	0	0	3	0	7	0	0	0	0	0	0	0	0	4	0	8	0	0
0	0	0	2	6	0	0	0	0	0	0	0	0	0	0	4	6	0	0
0	0	0	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Logioally Differenced Output

REPEAT.>2 20 100 0.99 0

9	-2
4	-3
1	-2
0	-2
2	-3
-1	-1
1	-4
1	-1
-1	-1
-2	3
-2	2
-4	0
-3	-4
-2	-1
3	12
2	4
1	1

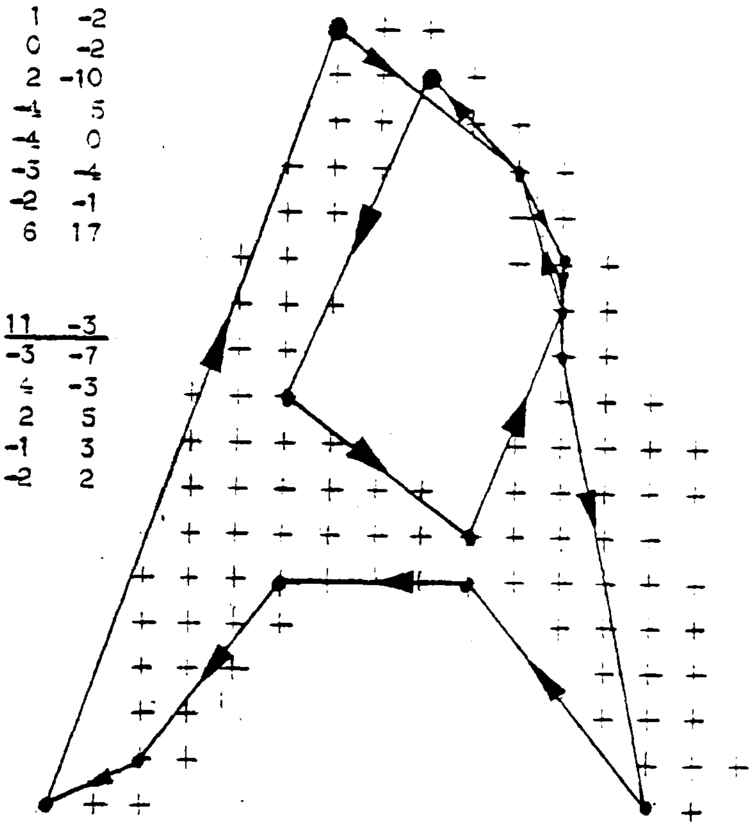
11	-3
-1	-2
-2	-2
1	-1
-1	-2
3	-2
1	-1
1	2
1	3
-1	3
-2	2



Dwait 2 20 100 0.9 1

9	-2
4	-3
1	-2
0	-2
2	-10
-4	5
-4	0
-3	-4
-2	-1
6	17

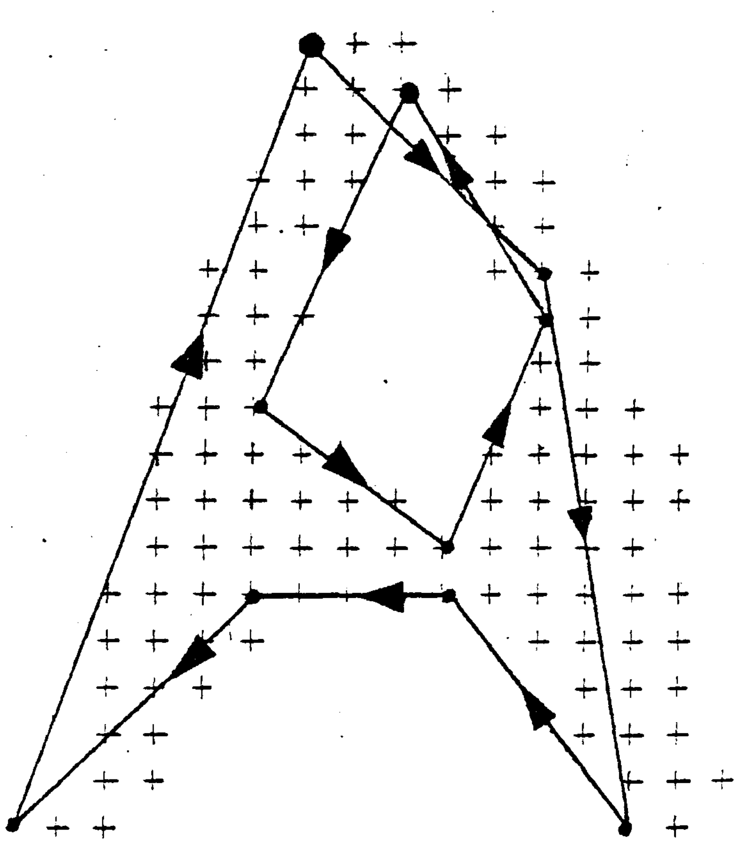
11	-3
-3	-7
4	-3
2	5
-1	3
-2	2



REPEAT.>2 20 100 0.8 1

9	-2
3	-5
2	-12
-4	5
-4	0
-5	-5
6	17

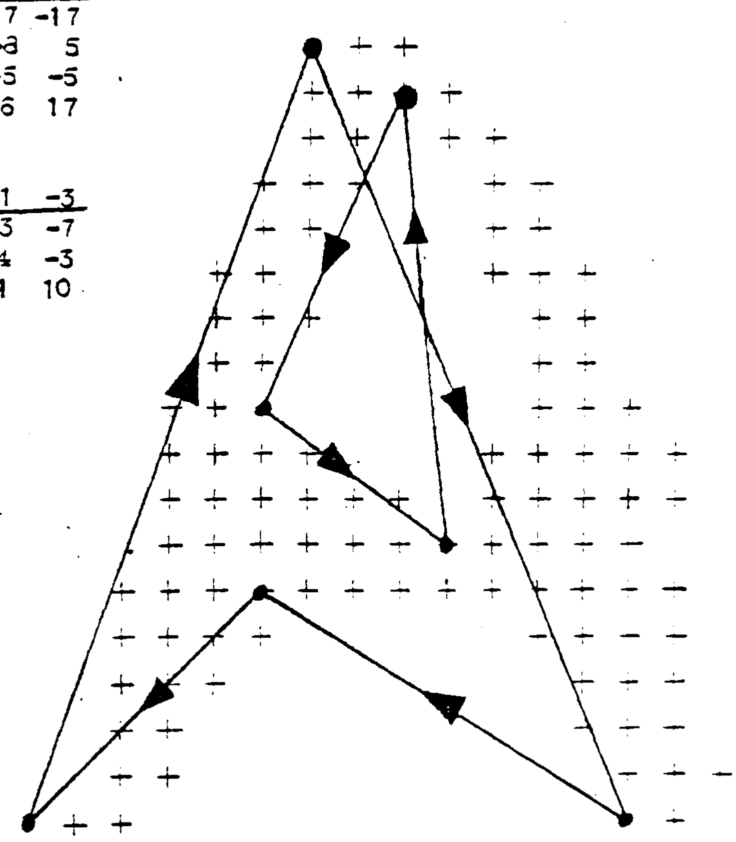
11	-3
-3	-7
4	-3
2	5
-3	5



REPEAT.>2 20 100 0.55 1

9	-2
7	-17
-8	5
-5	-5
6	17

11	-3
-3	-7
4	-3
-1	10



Vector Chain Output